Andi Rexha, MSc.

TITLE

Assessing the Usefulness of Propositions, Word Embeddings, and Style

to achieve the university degree of

Doctor of Philosophy

Doctoral's degree programme: Computer Sciences

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Denis Helic

Institute of Interactive and Advance Data Science
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Stefanie Lindstaedt

Graz, June 2020

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Abstract

Natural Language Processing (NLP) is an interdisciplinary field connecting computer science with linguistics. One way to classify NLP tasks is by the type of features we use to solve them. From that, we can distinguish between three high-level feature categories: semantic, syntax, and style. In this thesis, I show the usage of each of these features' categories in various tasks. The applications in these tasks assess the usefulness of propositions, word embeddings, and style.

In the first part of the thesis, I assess the usefulness of propositions (via Open Information Extraction (OpenIE)) using semantic features. More specifically, I study the application of these features for two sentiment analysis tasks: "aspect extraction" and "polarity determination". Usually, solutions for these tasks train models on specific datasets and achieve good results, but they fail to generalize in new contexts. This thesis proposes an unsupervised approach via OpenIE, which performs as good as top state-of-the-art systems by generalizing better in new data distributions. Such a result helps me answer the first research question of this thesis:

RQ1: Can a combination of Open Information Extraction and semantic features achieve state-of-the-art results on "aspect extraction" and "polarity estimation" for generic opinion streams?

After showing the effectiveness of OpenIE (thus prepositions) for sentiment analysis, I also study its use in a summarization scenario. Like the sentiment analysis case, the goal here is to monitor different news (social) media without knowing in advance the topic of their articles, hence favoring unsupervised approaches. More precisely, I propose a 4W block summarization approach for news (social) media monitoring. The output of this work has been used to summarize news (social) media for human operators and helps me answer my second research question:

RQ2: How do we adapt Open Information Extraction for generalized summarization in news (social) media monitoring?

Next, I continue studying another case of sentiment analysis, but this time by using syntactic features. To classify the polarity of aspect phrases in tweets, I use non-contextual word embeddings as features in a ”bag of words” approach. I also check the performance differences between the full tweet usage and a window of words around the aspect. The result of this work serves me to answer my third research question:

RQ3: To what extent does a ”bag of words” approach using word embeddings predict the sentiment in tweets, and what is the difference with just using a window of words around the aspect phrase?

My studies on syntactic features extend to contextual word embeddings. I apply them to short text classification cases with small datasets and compare the results to previous baselines. I show that in this scenario, the use of contextual embeddings outperforms previous baselines. Furthermore, I discover that enriching the dataset with these embeddings does not improve the classification performance, and sometimes even has a detrimental effect. This outcome helps me answer the fourth and fifth research questions:

RQ4: Is it possible to design a neural-based architecture that can build effective models from small datasets that outperform state-of-the-art data augmented techniques?

RQ5: Would the use of data-augmentation techniques in contextual word embeddings for text classification have a detrimental effect on the overall effectiveness or, at least, to have no significant improvements?

The last dimension investigated in this thesis is the writing style. Unlike previous works in authorship attribution, I present an algorithm that uses only stylistic features to identify the number of authors that have written scientific papers. Moreover, I also present pilot studies that suggest that, in some cases, it is impossible to identify the author of a piece of text just from their writing style. Both of these studies allow me to answer my sixth and seventh research questions:

RQ6: To which extent can we identify the number of authors of scientific papers just from their writing style?

RQ7: Is it possible, even for humans, to identify the authors in short text with high content similarity?

As part of an accumulative thesis, each of these research questions is backed by workshop, conference, or journal papers.

Contents

List of Figures

1 Introduction

Natural Language Processing (NLP) is the branch of Artificial Intelligence that connects computer science with linguistics. The main goal of NLP is to extract structured information from unstructured, naturally written text. Such extraction can vary from Text Classification (TC), Information Extraction (IE), Machine Translation (MT), etc. For an example of TC, one can think of a system that categorizes emails to ”spam” or ”not-spam”. IE’s goal is to annotate parts of the text and assign them to a class. An example of IE can be the identification of names of cities and villages expressed in a text. Finally, MT is the task of producing a translation of the text to another language (for example, Google Translate). There are different ways to classify the NLP tasks, but I want to focus on distinguishing the type of features used to solve them. At a high-level, this classification identifies three categories of characteristics of the language that one can exploit: semantic, syntax, and style.

Semantic represents the meaning and usually is expressed via structures that allow inference in which formal methods can be applied. Among these structures, one can mention dictionaries, thesaurus, lexical databases, sentiment lexicons, and ontologies. Dictionaries map the meaning of words in different languages, but in the NLP world, they also refer to lists of words that correspond to a specific context. Thesaurus maps a word to its synonyms (words or phrases that mean exactly or nearly the same) and antonyms (words opposite in meaning) within the same language. A more complicated semantic structure is the lexical database. Such databases extend thesaurus relations with hyponyms (words of a more specific meaning) and meronyms (constituents or parts of an object). Other types of semantic structures are sentiment lexicons that assign a polarity (whether it has a positive, negative, or neutral meaning) to each word or phrase. Finally, an ontology shows the properties of a subject area and how they are related, by

defining a set of concepts, categories, and relations that represent the subject. Ontologies are very helpful for sharing knowledge and automatic reasoning. Once they are defined (semi-automatically or manually), a reasoner can be applied to infer new logic relations between objects or concepts of the subject.

The syntax is about the surface content of the text. Usually, the textual resources are split into smaller pieces (words, sentences, paragraphs, etc.), and each of these pieces is then analyzed and used as features for the downstream task. In this category, there are two ways of extracting features. One is via preprocessing tools such as morphological or grammatical preprocessing, and the other is via embeddings. The morphological preprocessing uses tools like stemmers or lemmatizers, which transform the inflections of words (for example, plural and singular) to shared content. The grammatical analysis parses the text usually at a sentence level, by assigning relations between the words. One of these relations is Part of Speech (PoS) tagging, which attaches to each word a tag such as Noun, Proper Noun, etc. Chunking is another grammatical parser which groups words within tag-phrases such as Noun Phrases, Verbal Phrases, etc. Another heavily used analyzer is the dependency parsing, which creates a graph structure with tagged dependencies between words.

A different way of extracting information from text is to assign to each of the words a vector that reflects a learned hidden state, called embeddings[1]. Word embeddings have proven to be very useful in NLP tasks, especially in machine learning.

The style is the ability of a person to write in different manners from others. It is a form of a ”fingerprint” that helps us differentiate authors. It is challenging to decouple the style from content, as they are strictly related. However, some writing characteristics distinguish us, for example, from “prepositions” one uses or the length of the sentences that one writes.

The previously explained categories are often combined. Indeed, information from semantic resources is often used to enrich features extracted in

[1]One can argue that word embeddings are not syntactic features. This is an open debate, as word embeddings contain both syntactic and semantic information. In this work, I consider word embeddings as mainly syntactic features.

the syntactic analysis and sometimes as a combination with style for authorship attribution. The syntax is also useful for authorship attribution and is sometimes used to extract semantic resources automatically.

There are two general approaches used in NLP tasks, depending on their goals. If the goal is to infer new relations on a semantic resource, the system is built via semantic reasoning by exploiting the formal language's power. If the task deals with unstructured text, usually features are extracted and used in machine learning methods. Machine learning methods can be categorized into different types, but the most used ones in NLP are supervised and unsupervised methods. Supervised methods find rules (or weights) of the features that can predict an instance's class (or an associated numerical value). Unsupervised approaches do not rely on label data; instead, they try to identify documents' common patterns.

This thesis's contribution touches the three categories of features previously mentioned in both manners, supervised and unsupervised. I first start assessing the usefulness of propositions with the help of Open Information Extraction (OpenIE). OpenIE is an NLP task that extracts propositions (the most basic meaning of a statement) of the form {*subject; predicate; object*} from sentences. In my work, I start exploring the amount of information that one can obtain from such a construct by using it in two different scenarios: 1) a sentiment analysis and later 2) in a summarization scenario. For the sentiment analysis case, I show the extraction of the aspect (the target of the opinion) from propositions and use the polarity of the words and phrases with sentiment lexicons. Thus I show a combination of the semantic with the syntax for extracting features. With this work, I answer my first research question:

RQ1: Can a combination of Open Information Extraction and semantic features achieve state-of-the-art results on ”aspect extraction” and ”polarity estimation” for generic opinion streams?

The second scenario that I present here is summarization, where I use OpenIE propositions (thus syntactic features) for ranking and showing information blocks to human operators. In this study, I show an unsupervised summary extraction from news (social) media that answers the following research questions:

RQ2: How do we adapt Open Information Extraction for generalized summarization in news (social) media monitoring?

I shift to assess the usefulness of syntactic features in sentiment analysis from unsupervised with the use of semantic resources to supervised with the use of syntactic features. More specifically, I have explored word embeddings, such as Word2Vec, as features for sentiment analysis to answer the following research question:

RQ3: To what extent does a "bag of words" approach using word embeddings predict the sentiment in tweets, and what is the difference with just using a window of words around the aspect phrase?

To extend the assessment of the knowledge contained in word embeddings, I use contextual word embeddings such as BERT to show the difference in classification compared to the usage of just surface words, in short-text classification with a low amount of training data. Furthermore, I explore the possibility of enriching the dataset with BERT's ability to identify similar words in the context. This thread answers the following research questions:

RQ4: Is it possible to design a neural-based architecture that can build effective models from small datasets that outperform state-of-the-art data augmented techniques?

RQ5: Would the use of data-augmentation techniques in contextual word embeddings for text classification have a detrimental effect on the overall effectiveness or, at least, to have no significant improvements?

Another way to try the supervised approach is with a different kind of features, namely stylistic features. To assess the usefulness of the writing style features, I check the text's difference written by multiple authors. I predict the number of authors in such documents, solely based on the way they write. Finally, I show the possible limitations of stylistic features with parts written by different authors, that have a very high content similarity. This line of research helps to answer the following questions:

RQ6: To which extent can we identify the number of authors of scientific papers just from their writing style?

RQ7: Is it possible, even for humans, to identify the authors in short text with high content similarity?

To give an overview of the scientific dissemination from this research, in Figure 1.1 I show a list of publications that I have authored (or co-authored). Each paper is mapped to the main category of features used to solve the tasks. The dotted framed papers are not presented in this thesis as they do not answer the hypothesis I want to address and defend in this thesis.

To summarize, the papers presented here with the corresponding research are shown below:

- ReUS: a Real-time Unsupervised System For Monitoring Opinion Streams
- Social Media Monitoring for Companies: A 4W Summarization Approach
- Polarity Classification for Target Phrases in Tweets: a Word2Vec Approach
- A Neural-based Architecture For Small Datasets Classification
- Towards Authorship Attribution for Bibliometrics using Stylometric Features
- Extending Scientific Literature Search by Including the Author's Writing Style
- Authorship Identification of Documents with High Content Similarity

In the following sections, I detail each of these research questions addressed here.

1.1 Assessing the Usefulness of Propositions

Information Extraction (IE) is the NLP task of extracting or annotating specific information from a predefined domain, for example, identify in a plain text the names of restaurants, email addresses, etc. Knowing the areas of interest in advance is not always possible. If one wants to build an Information Retrieval System for the Web, she cannot identify the eventual user requests in advance. Yates et al., 2007 proposed the Open Information

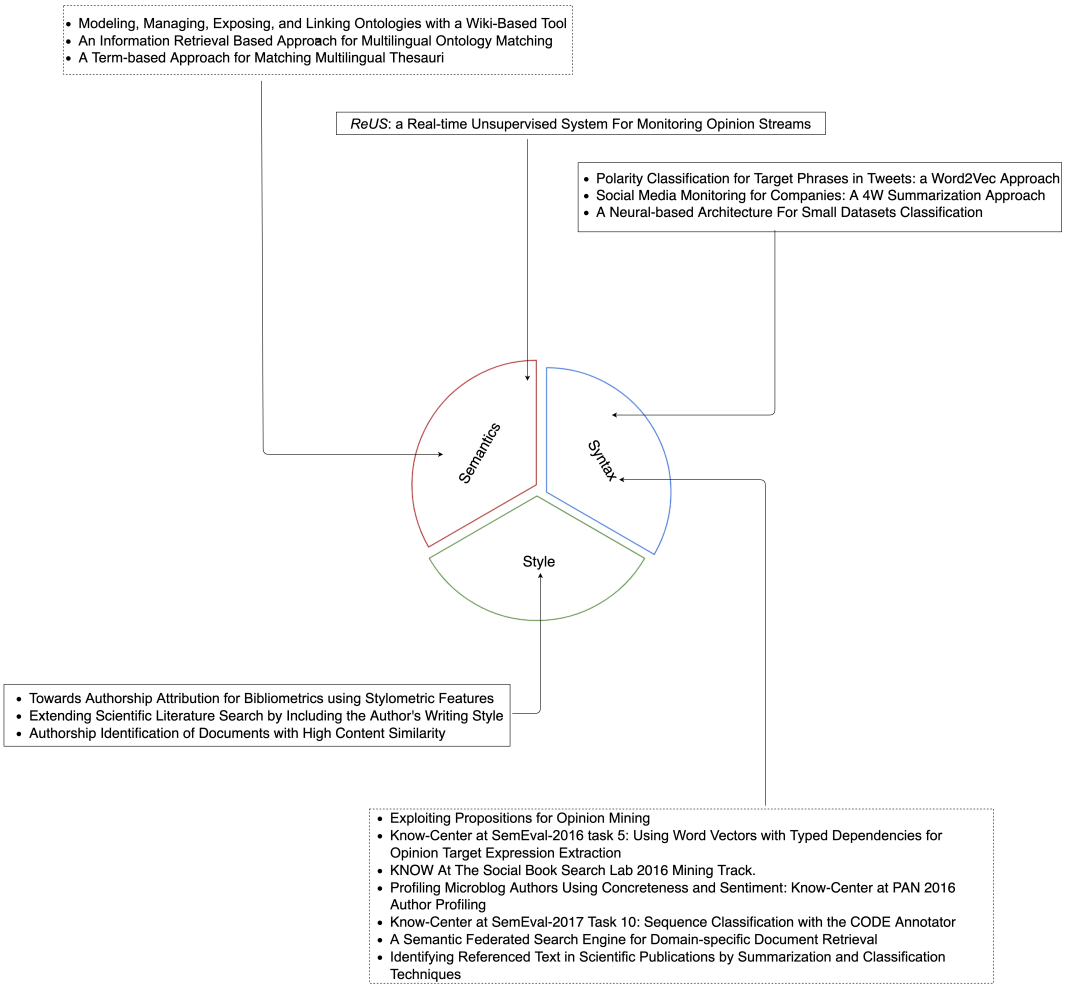


Figure 1.1: The mapping to each dimension of the publications I have coauthored. The titles on a dotted frame represent papers which I am not going to present here. The titles framed in a filled line, represent the papers of my PhD thesis. For these publications I have collaborated with the Know-Center, TU Graz and Fondazione Bruno Kessler (Trento)

Extraction (OpenIE) paradigm, which extracts a broad set of relational tuples called propositions, without knowing them apriori. Typically, the relation name is just the text linking two arguments; for example, in the sentence "Donald Trump was born in New York", OpenIE should identify the tuple {*Donald Trump; born in; New York*}, which has the relation *"born-in"*.

Apart from the definition of the OpenIE paradigm, Yates et al., 2007 introduced also the first system called *TextRunner*. The authors use a set of heuristic from a dependency parsing output in a small dataset to produce positive and negative examples. These examples are then parsed with a PoS tagger, and the tags are used as features for a naïve Bayes classifier. To note here that the authors explicitly avoid the use of dependency parsing for extracting the features of the final model for efficiency reasons. Indeed, the dependency parser has a much longer execution time than shallow parsing. Different works after *TextRunner*, have used similar approaches. WOE^{pos} from F. Wu and Weld, 2010, and *ReVerb* from Fader, Soderland, and Etzioni, 2011 use supervised and semisupervised approaches on the lexical and shallow parsing of each sentence. The advantage of these systems relies on the speed of tuple extractions due to the shallow preprocessing nature. On the other hand, they produce wrong propositions, and most importantly, they do not find all of them.

To improve extraction quality, later generation systems started using deep paring outputs as features, with Akbik and Löser, 2012 proposing *Kraken*, Mausam et al., 2012 proposing *Ollie*, Del Corro and Gemulla, 2013 proposing *ClausIE*, and Bast and Haussmann, 2013 proposing *CSD-IE*. These systems use rule-based, supervised, and semisupervised approaches but built on more complex features than the previous ones.

The newest solutions are based on Deep Learning (DL) approaches. To model the extraction of the OpenIE tuples, B. Kim, H. Yu, and G. G. Lee, 2016 use LSTMs (Long Short Term Memory, a deep learning architecture), and Cui, F. Wei, and Zhou, 2018 uses an encoder-decoder architecture, both trained with data collected by previously built systems. Such systems prove a better efficiency and quality robustness in the presence of noise than previous generation systems.

OpenIE has been explored mainly in IR systems as one can search for "actions" (relations). A query of the form "who was *'born in'* New York" can easily be parsed and matched to the extracted tuples with "*born in*" as relation and "New York" as an object. An IR system would return subjects of the extracted tuples. In this thesis, I have studied the use of OpenIE in two new scenarios: 1) Sentiment analysis and 2) Summarization.

OpenIE for Sentiment Analysis Opinion mining and sentiment analysis are NLP tasks that aims to extract opinions from texts, classify them with a value representing the overall polarity (*positive*, *negative*, or *neutral*) associated with a given subject (Pang, L. Lee, and Vaithyanathan, 2002; Cambria, 2016). An example of these tasks can be found in social media, where users express their opinion about different topics and products. Such a scenario has been studied initially by Go, Bhayani, and L. Huang, 2009; and Barbosa and Feng, 2010. These systems assign a total polarity value to the whole text. This strategy is not viable if the goal is to identify different facets of opinions. Consider the example: "*The overall experience was good, but the instructor was really bad*". If one assigns a polarity value to the whole opinion, it will lose information, regardless of the returned polarity value. Indeed, it would be ideal for extracting the positive polarity on the "*experience*" and the negative polarity of the "*instructor*". The goal here is first to identify such textual segments, so-called *aspects*, and then extract their correspondent polarities (Hu and B. Liu, 2004).

From the example shown above, it seems quite natural the association between propositions from OpenIE and the identification of the aspect with its polarity. Indeed, a good OpenIE system would produce the tuples {*experience, was, good*}; {*instructor, was, really bad*}. These tuples help in identifying the aspect (each aspect resides in a different tuple) and relate the words that associate the polarities to them.

Most works on aspect extraction base their solutions on supervised methods applied to lexical features. These approaches use sequential classifiers to identify the aspects of opinions. In (Jakob and Gurevych, 2010) authors used conditional random fields (CRF), while (Jin, Ho, and Srihari, 2009) authors used hidden Markov models (HMM) on lexical features and Part of Speech (PoS) tags. (B. Liu, Hu, and Cheng, 2005) used sequential rule mining on

the same features and (Y. Wu et al., 2009) used additional dependency trees features. Recent supervised approaches use sequential classifiers and deep learning, mainly on word embeddings. Among these works, I identified papers such as (W. Wang et al., 2016; Xu et al., 2018; and He et al., 2018).

The unsupervised methods fit models on specific datasets, and although they achieve excellent results, they fail in generalizing for new contexts. Our work proposes a generic unsupervised approach that does not overfit specific datasets and performs similarly to supervised techniques. Other unsupervised approaches have been proposed in the literature and are mainly based on topic modeling. Topic modeling is a statistical method that uses the co-occurrence of words in documents to identify topics. Among this line of research we can mention (Mei et al., 2007; Titov and McDonald, 2008; Li, M. Huang, and Zhu, 2010; Mukherjee and B. Liu, 2012; and Y. Wu et al., 2009). Here the authors use the topics as aspects for opinion mining but do not achieve results as good as supervised methods.

In this thesis, I present a solution to extract the aspects of opinions and assign them a polarity value. Given a review written in social media, the first step is to obtain its propositions. From them, we try to identify whether there is an aspect to be considered. Later, we use the dependency graph to apply a set of rules that extract a list of relationships "aspect $\leftarrow$ {set of relations}". The polarity assigned to the aspect is the average polarity of the set of relations within the following sentiment lexicons:

- *SenticNet*[2] is a publicly available resource for opinion mining that exploits both artificial intelligence and semantic Web techniques to infer the polarities associated with common-sense concepts.
- *General Inquirer*[3] is an English-language dictionary containing almost 12,000 elements associated with their polarities in different contexts.
- *MPQA*[4] is a sentiment lexicon built for Multi-Perspective Question Answering purposes.

We have evaluated the study in two datasets from the challenges *SemEval 2015 Task 12*, and *SemEval 2016 Task 5*. These challenges consist of identifying

[2] https://sentic.net/
[3] http://www.wjh.harvard.edu/~inquirer/
[4] https://mpqa.cs.pitt.edu/

the aspects of reviews from different domains, such as **restaurants**, **hotels**, and **laptops**. Even without fitting datasets' specific features to our system, we perform among the top teams on these challenges. Our approach achieves second place in extracting the aspect from a **restaurant** domain with an F-Measure of 0.6036 for *SemEval 2015*. The winner of such competition made 0.6268 without performing significantly better than our system. In the same challenge's dataset, for the **laptop** aspect detection, our method shows better results than every other system with an F-Measure of 0.5131. In the *SemEval 2016 Task 5*, we perform statistically significantly better than every other team for **laptops** with an F-Measure of 0.5692. Still, it achieves the third-place of 0.6687 F-Measure for the restaurant domain, which is statistically significantly worse than the first two positions (with the winner being at 0.7234).

As for the polarity identification, the challenges consist in classifying the review between "positive", "negative", and "neutral". We statistically outperformed the other systems for the **laptop** domain with an accuracy of 0.8589 for 2015 and 0.8710 for 2016. In the domains of **restaurants** and **hotels** we achieved for both years a second place. In *SemEval 2015* we riched a 0.7794 (compared to the 0.7869) and 0.8524 (compared to the winner with 0.8584) of accuracy respectively. For the *Semeval 2016* the result was 0.8710 (compared to 0.8813) for **restaurants** and 0.8710 (compared to 0.8813) **hotels** 0.8524 (compared to 0.8584 of the winner). In both cases, the winners did not perform statistically significantly better than our results.

Even without using a supervised solution based on the training data's content, we perform among the best teams of the *SemEval*, where the challenge deals with different contexts for aspect extraction and polarity detection. These results help me answer my first research question:

RQ1: Can a combination of Open Information Extraction and semantic features achieve state-of-the-art results on "aspect extraction" and "polarity estimation" for generic opinion streams?

OpenIE for Summarization After showing the usefulness of propositions in sentiment analysis, I extend the study of OpenIE in a different scenario,

that of summarization. Text summarization refers to techniques of compressing the length of information by maintaining most of its semantics. Summarization systems display to the user diverse types of information, such as keywords or sentences. These systems use various techniques to extract such information, but one can classify these techniques into two categories: extractive and abstractive. In extractive summarization, the displayed summary is a substring of the text. An example of these systems can be (Erkan and Radev, 2004), where the summary sentences are selected based on their similarity with each other. Other extractive approaches model the text as a graph and apply the PageRank algorithm to identify the most prominent sentences (Mihalcea and Tarau, 2004, and a further improvement from Seifert et al., 2013). Gillick and Favre, 2009 propose an Integer Linear Program (ILP) to exact inference under a maximum coverage model for automatic summarization. More recent approaches apply neural networks to learn to select the most representative sentences (Cao et al., 2015; Alon, Levy, and Yahav, 2018; and Fernandes, Allamanis, and Brockschmidt, 2018).

Abstractive summarization is not using substrings from the text to display the summary. Instead, the systems try to detect the main underlying concepts and later generate a summary by using text synthesis. Examples for abstractive summarization are the based on sentence compression (Knight and Marcu, 2002) and the use of semantic graphs (F. Liu et al., 2018). Recently, the use of the transformer-based architectures has created new abstractive summarization systems. Systems such as Bart (Lewis et al., 2019) or T5 (Raffel et al., 2020) are pre-trained on generic datasets, and they adapt quite well in this task.

Here, I want to present an OpenIE summarization approach for news (social) media. This approach can help human operators to avoid reading the whole stream of data by showing them only an easy to grasp summary. What we noticed is that temporal and positional expressions are very informative. Furthermore, an intuitive way to describe events would have the form of ”who did what”. Thus, we propose to display a list of ”event” blocks to the user that answers the questions: ”Who?”, ”What?”, ”When?” and ”Where?”; called a 4W summarization approach. In this approach, the events are modeled naturally as propositions. To illustrate this method, consider the following example:

"Coronavirus surfaced in a Chinese seafood market two months ago."

The extracted block would be:

Who: "Coronavirus";
What: "surfaced",
Where: "in a Chinese seafood market",
When: "two months ago".

To extract these blocks, we map the "Who?" to the "subject" and the "What?" to the "predicate" of propositions. The "Where?" and "When?" are parts of the object. To extract such elements, I first annotate the sentence with Temporal and Named Entities expressions. The "When?" is derived from the temporal expression in the object, and the "Where" from Geographical Named Entities and a list of heuristics.

As each sentence might have multiple propositions, I explore three different strategies for selecting the 4W blocks from a sentence:

- The longest proposition: that contains the largest number of words.
- The shortest proposition: that does not lose Named Entities and Temporal Information.
- The shortest proposition: st selects the proposition which contains the smallest number of words.

The next step of the summarization strategy is about selecting the 4W blocks that better represent the whole text. To do so, we select four different summarization strategies:

- First/last: selects the 4W blocks from the first sentences and the last one. (inspired by the Lead method from Baxendale, 1958).
- Coverage: selects 4W blocks with the most question types answered.
- Up-to-dateness: selects 4W blocks where the corresponding sentence's tense is either the present or future tense.
- TextSentenceRank: an existing extractive summarization algorithm.

We evaluated such an approach and achieved a 0.343 F1-Measure for the selected sentences via the Lead strategy. We also achieved a 0.932; 0.861; 0.900; 0.803 F1-Measure respectively for "Who?", "What?", "When?", "Where?" questions. This results help me answer my second research question:

RQ2: How do we adapt Open Information Extraction for generalized summarization in news (social) media monitoring?

After showing the usefulness of OpenIE for summarization and sentiment analysis, I start investigating the latter task by using syntactic features. To do so, I employ word embeddings also in a classification scenario. The next section shows my studies on this topic.

1.2 Assessing the Usefulness of Word Embeddings

In the previous section, I have shown an unsupervised scenario for sentiment analysis that uses features extracted from semantic lexicons to identify the polarity. In this section, I start to describe my work for the same task, though on a different dimension, which means by using a supervised approach with syntactic features.

Traditionally, words or terms[5] are extracted from the text and used as syntactic features in their surface form. Sometimes, the stems, lemmas, or word n-grams (n-words occurring next to each other) extend the feature space. These features are then placed in a vector where each element corresponds to a surface form in the vocabulary. In most of the cases, a machine learning algorithm is employed to learn the task by applying rules or assigning weights to each of the features. One issue with these features is their high dimensionality and sparsity, known as the curse of dimensionality. Synonyms and other words with the same semantic but different surface forms are considered distinct. We want to match them somehow so that the learning algorithm can detect patterns of similar terms. One way to overcome the curse of dimensionality is by using lexical databases such as WordNet (Fellbaum, 1998). The synonyms improve the matching of the terms, but cannot fully solve the problem. Indeed, most of the words will still be distinct from each other, even though they might have a related semantic similarity.

The need for the density of feature space led to new methods for feature encoding. One proposed way in the literature is via topic modeling. Topic modeling is an unsupervised machine learning technique that learns

[5] In NLP "words" and "terms" are usually used as synonyms

abstract topics for documents. One of these methods is the Latent Semantic Analysis (LSA from Landauer, Foltz, and Laham, 1998). LSA builds a matrix containing the occurrence of terms per document. Rows represent the unique words, and columns represent documents. Singular Value Decomposition (SVD) is then applied to the matrix to lower the ranks. As a result, the algorithm produces dense representations for each document and term, and it is used to identify similar terms and documents. LSA is also applied in information retrieval for identifying synonyms called Latent Semantic Indexing (LSI from Deerwester et al., 1990). Other techniques, such as Probabilistic LSA (pLSA from Hofmann, 2013) and Latent Dirichlet Allocation (LDA from Blei, Ng, and Jordan, 2003), use probabilistic methods to decompose and lower the ranks of the matrix. The use of topic modeling approaches gives us a dense representation of words at a document level, but they have not shown outstanding NLP tasks' performance.

Dense representations for words that perform quite well in NLP tasks are word embeddings. The most common way to learn them is by training large generic corpora on Language Models (LM). Language Models try to predict missing or future tokens given a context. One example of LM is identifying from a sentence a missing token given the rest of them.

The first embedding systems produced a single representation for each word, independently of the context. Such representations contain an analogy effect where words with similar meanings have the same distance in space (for example, the gap between the words "king" and "man" is approximately the same as the one between "queen" and "woman"). Among the systems of this category, which I call non-contextual, one can find Word2Vec (Mikolov et al., 2013), GloVe (Pennington, Socher, and Manning, 2014), fastText (Joulin et al., 2016), etc.. In this thesis, I start testing the effectiveness of the embeddings extracted from these systems for sentiment detection in tweets. This work helps me answer my third research question, as explained later in this section.

The second family of word embeddings is the contextual one. Contextual word embeddings do not produce a single representation for each word but one that depends on the context. Among these systems, we can find BERT (Devlin et al., 2019), ELMo (Peters et al., 2018), RoBERTa (Y. Liu et al., 2019), etc.. In this work, I use the BERT system in a supervised scenario and

compare it with classic supervised methods applied to features from the surface form of the words. I further demonstrate that differently from the surface form features, enriching contextual word embeddings does not have any positive effect in short text classification. These results help me answer my fourth and fifth research questions.

Polarity with Word2Vec To check the effectiveness of word embeddings, I start investigating the task of sentiment analysis on twitter. Previous to my work, other authors have conducted studies for twitter classification. Go, Bhayani, and L. Huang, 2009; and Barbosa and Feng, 2010 use both emoticons and words as features with the latter, additionally using word n-grams and PoS tags. Agarwal, Biadsy, and Mckeown, 2009 classify the sentiment of the tweet via a dictionary of emotions called DAL (Dictionary of Affect in Language).

In this work, we assess the usefulness of word embeddings on twitter. We start by checking the performance of different classifiers for polarity detection. This approach mimics a form of ”bag of words” model, where the words' order is not considered. For this, we extract for each token of the tweet, its Word2Vec representation, and enrich some handcrafted features. The embeddings are then averaged and input to four different classifiers: Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Random Forrest (RF).

After using the whole tweet, we also try a different configuration, where we consider only a window of three tokens around the aspect phrase. This configuration aims at identifying the polarity effect of distant words from the aspect. The results showed the best outcome was achieved from SVM with a 0.896, 0.536 (for positive and negative polarity respectively) for the whole tweet and 0.855 and 0.429 (respectively) with features from a window close to the aspect. These outcomes suggested that we generally lose classification accuracy by using only a window of terms close to the aspect. This study helped me answering my third research question in this thesis:

RQ3: To what extent does a ”bag of words” approach using word embeddings predict the sentiment in tweets, and what is the difference with just using a window of words around the aspect phrase?

The early embeddings like Word2Vec are very powerful and have proven to work quite well in different NLP tasks, but sometimes lack in the contextual representation of the word itself. For example, the word *bank* can have different senses, such as a *riverbank* or a *financial institution*. This ambiguity may mislead systems built for NLP tasks that are trained on the embeddings of the words. To overcome such ambiguity, and extend the use of word embeddings for classification purposes, I have conducted a different study with the current generation of word embeddings, the contextual ones.

Classification and encrichment with BERT BERT is a contextual embedding system with which I conduct my experiments and is one of the most used word embeddings recently. Its architecture is formed by a Neural Network based only on attention mechanisms. The training of such network goes through two steps:

- Language Model: Words are hidden (masked) with a probability of 15%. The goal of the network during this phase is to identify missing words.
- Sentence Entailment: The system is trained with two sentences following each other, and the goal is to identify whether the second follows the first in any text.

The first step was devised for learning the embeddings and the second for fine-tuning them. Indeed, the Language Model task learns very good embeddings and generalizes in many NLP tasks, as proven in most previous research. While the Sentence Entailment fine-tunes the systems for tasks such as Question Answering (QA).

In the current study, I focus on the classification of short text with a small amount of data available. More precisely, I have used BERT embeddings two folds:

- To propose a new baseline for short text classification.
- To show that the text enrichment with word embeddings might even have detrimental effects on the classification models.

Authors have widely researched the small dataset classification problem. Toutanova et al., 2001 was one of the early works in this scenario. The

approach used in this study exploits a Bayesian classifier based on word features in a hierarchy of classes. In another study, Shridhar et al., 2019 apply Semantic Hashing (a form of embedding without continuous values) for Intent Classification. Clarizia et al., 2011 used LDA (Latent Dirichlet Allocation) to create a model where documents are associated with their topics via probability value. These probabilities are then used as features for small dataset classification. K. Kim et al., 2019 propose a novel architecture, which trains a Convolutional Neural Network (CNN) in two datasets, a small and a large one, as two related tasks. The shared features help to improve the accuracy of the task with a small number of instances.

To improve the accuracy of classification in small datasets, authors propose different methods to augment the classification instances. Lu et al., 2006 and Abulaish and Sah, 2019 uses Latent Dirichlet Allocation (LDA) as means of text augmentation. The former uses LDA as enrichment the same way as the embeddings. The latter uses LDA to extract keywords that are contained only in one class. Each instance of that class is then enriched with the keywords. J. W. Wei and Zou, 2019 propose a simple method for data augmentation by randomly selecting words to substitute with their synonyms or delete them. New instances for training are generated from such operation. In S. Yu et al., 2019, the authors propose to stack two levels of bidirectional Gated Recurrent Unit (GRU from Cho et al., 2014) with attention layers in between. The system first runs the whole dataset on the network, by training for the classification task. Next, the attention layers are used to select the most important words of the sentences. These words are then used as a selector for the augmented document. Rizos, Hemker, and Schuller, 2019 proposes another method for increasing the number of instances in the training set. In this method, words are substituted with synonyms that have their same part of speech. Next, the authors train a Language Model via a Long Short Term Memory (LSTM from Hochreiter and Schmidhuber, 1997) network with the current dataset. A randomly selected word is used as inception for the creation of a new instance, with the rest predicted from the LSTM. Synonyms for generating new instances are also used in Lochter et al., 2018. Schulz et al., 2016 and Veliz, Clercq, and Hoste, 2019 try to increase the number of instances via spelling error fixes.

My work in this thesis for short text classification with a small dataset

is based on two previously introduced baselines. The first baseline was proposed by S. I. Wang and Manning, 2012 and used words and words bi-grams as features with two different algorithms, namely Multinomial Naive Bayes (MNB), the Naive Bayes Support Vector Machine (NBSVM). Elekes et al., 2019 extends these baselines with the Recursive Auto-Encoder (RAE) algorithm. The authors use these algorithms on four short-text datasets benchmarked in numerous NLP studies: Customer Reviews (CR), MPQA Opinion Corpus, Short Movie Reviews (Rt10k), and Subjectivity (Subj). To analyze small datasets with short texts, they use different training-set sizes: 500, 1000, 1500, 8500 for MPQA, Rt10k and Subj, 500, 1000, 1500, and 2600 for CR. From the baselines with these datasets, an enrichment method is proposed. The intuition behind this enrichment is to use synonyms that are specific for the task at hand. The authors first cluster terms from their Word2Vec representation and later check whether the words have a similar distribution (are generated from with the same stochastic process) and if so, they can be substituted. The substitution creates a new instance that is used to grow the dataset. The authors show an improvement compared to the baseline classification.

In our study, we first check how good a BERT based classifier works compared to the current baselines suggested by Elekes et al., 2019, and their enriched version. The nature of the datasets used by the authors is binary, which means that the instances have to be classified between two classes. Thus, in the BERT architecture, we add a final sigmoid node as it adapts very well to the data's binary nature. After training and testing, results show that our architecture outperforms both the baseline and the enriched training dataset, by de facto creating a new baseline for short text classification with a small amount of data. As further research, we start checking the effect of the dataset's enrichment. As mentioned previously, BERT's first phase training does exactly what we want from text augmentation. The algorithm should detect substitutes that make the sentence "naturally" written by masking words from a sentence. This hint led to an approach for data enrichment by first selecting the most prominent terms to be replaced with a TF-IDF (Term Frequency- Inverse Document Frequency) policy. This policy helps identify the most discriminating words of a document, therefore assisting us in avoiding irrelevant contextual words for replacement. Later, we try three different approaches for the substitution: 1) all the replacements from BERT,

2) the only replacements that are also synonyms (checked with WordNet), 3) only replacements that do not appear in the other class as replacements or as words. Running the same architecture on the enriched data showed that there was almost no effect or sometimes even detrimental the classification results. These outcomes helped me answering my fourth and fifth research questions:

RQ4: Is it possible to design a neural-based architecture that can build effective models from small datasets that outperform state-of-the-art data augmented techniques?

RQ5: Would the use of data-augmentation techniques in contextual word embeddings for text classification have a detrimental effect on the overall effectiveness or, at least, to have no significant improvements?

The last category of features not touched yet in this thesis is the writing style. Unlike the previously mentioned experiments, we need longer text to analyze the writing style than the current setup. In the next section, I provide the results of my work with this type of feature.

1.3 Assessing the Usefulness of the Writing Style

The writing style is one of the dimensions in which one can project NLP characteristics. The idea behind writing style is that each author has kind of a "fingerprint" in the way they write. Intuitively, when we read a report at work, we can sometimes identify if a text is written by one or another of our colleagues. Research on this topic dates back to the 19th century when Mendenhall, 1887 published the quantitative analysis of writing style. The goal of this work was to identify that William Shakespeare was not the real author of books attributed to him. To do so, Mendenhall used the distribution of words of various lengths. Writing style was also used to analyze the famous "The Federalist Papers" by Mosteller and Wallace, 1964, where they used features such as articles and prepositions to identify the real authors. These publications led to the research area known as stylometry. Stylometry focusses on identifying features that distinguish the authors from each other. Holmes, 1998 defines a list of stylistic features

extracted from the lexicon, syntax, or structure. The lexical stylistic features are derived from the vocabulary used in the text, such as average word length, vocabulary richness, etc. Syntactical style features are based on the way sentences are written, such as the frequency of prepositions and use of punctuation. Structural features are more about how the text is organized, for example, the number of paragraphs used, the indentation, or whether the author use or not space before a parenthesis.

As suggested in early works, the main goal of stylometry is authorship attribution. In studies like Juola et al., 2008, the task is to identify the author of an article from a given list. The approach followed by this study first selects stylistic features from each author and creates a supervised model with the output of the name of the author. A list of useful stylometric features and approaches for this task is compiled in Stamatatos, 2009. The features used are very dependent on the task at hand. Indeed, for short messages such as in Villar-Rodriguez et al., 2016; Brocardo et al., 2013 the importance of the features differ from authorship attribution in long unstructured texts (Zhang et al., 2014). Another application of stylometry is plagiarism detection (Culwin and Lancaster, 2001; Zechner et al., 2009; Stein, Koppel, and Stamatatos, 2007). This task's goal is to identify a written sequence that is copied and adapted by another author.

Although the previous works on authorship attribution are based on stylistic features, they also use syntactic features. By identifying trigger words for authors, one can, in some instances, create a simple system for authorship attribution. Consider, for example, an author writing about cars and another one writing about politics. Even though it might be trivial to build a system that differentiates these two authors (for example, the first one might mention brands of cars in her articles), the model does not generalize. If the first author starts writing about politics, the system will likely fail to recognize an article written by her. Unlike previous research, in this work, I focus only on the lexicon and structural stylistic features. Specifically, I study stylistic features in research publications with the primary purpose of helping information retrieval systems specializing in scholarly articles. The goal here is to avoid assigning group expertise to an author just because she co-wrote a paper. Indeed, idealistically, we would attribute each written segment of the text to an author. By doing so, we can identify the scientific expertise of single authors, which can help two-fold: 1) to identify subfield

experts, and 2) to credit authors differently. Intending to assign text passages to each author, in this thesis, I go through a list of publications that help to understand how much of it we can achieve. More precisely, I investigate the amount of information that we can extract from the style for authorship attribution.

Before starting the description of my studies in the field, I want to begin by describing an algorithm that I extensively used here. This algorithm, named *TextSegFault* (Kern and Granitzer, 2009), is used to identify changes in topics. To do so, it tries to identify the difference between two adjacent segments of text to each other, by using a similarity (dissimilarity) measure. More specifically, sentences are compared within and between the segments via this measure, and a final dissimilarity score is calculated. This score reports the difference between the two segments depending on the features one uses to encode the dissimilarity. Intuitively, this approach checks whether there is a discordance between the two segments, which usually indicates a change of topic. Finally, the algorithm slides over the positions of the text to find these topic changes.

Visualizing topic changes The first step toward my research topic is to distinguish between scientific papers written by different authors. In our first preliminary study in this field, we suggest a pipeline to analyze scientific articles by first extracting the structured text from the PDF file by using the work from Klampfl et al., 2014. We apply TextSegFault to the extracted text, but unlike previous studies, our method does not use syntactic features but only stylistic ones. These features are combined to define the dissimilarity between two segments. Indeed, we use cosine similarity between the two normalized stylistic vectors as a hyperparameter to the TextSegFault. To produce a preliminary evaluation, we decided to visualize how the style changes in scientific articles. For each document, we calculate all the dissimilarity values returned from TextSegFault. We plot the graph of style change in four different categories, papers written by one, two, three, and four authors. This visualization reinforces the intuition that the higher the number of writers, the more likely it is to have bigger topic changes. Surely, in some of the plots, there is an evident change in writing style.

Authorship attribution To make a quantitative study, we extend the first one, by selecting 6144 research articles in total, across 563 different journals from Pubmed written by one(983), two(1192), three(1391), four(1418), and five authors(1160). After normalizing the length of the documents, we applied two different supervised algorithms, Logistic Regression, and Random Forest. The first performs quite poorly, by achieving an average 0.362 of F1-Measure, while Random Forest performs 0.755 of F1-Measure both validated on 10-fold cross-validation. The main observation from the results is that we can distinguish at least three out of four times the amount of authors of a paper. By detail viewing our results, we discovered that our system performs worse to identify articles written by five authors. Our central intuition to explain such an outcome is that the larger the number of authors, the more likely it is that one of them didn't contribute to the paper. This result helps me answer my sixth research question:

RQ6: To which extent can we identify the number of authors of scientific papers just from their writing style?

High content similarity In the last work about stylistic features, I investigate whether it is possible in every case to distinguish authors just from their writing style. To do so, we have created a case study involving human judges. We first retrieved papers written by single authors and selected the introduction section's first sentences. By doing so, we have generic text snippets, as the beginning of the introduction is usually more generic than other parts of the text, each authored by an only person. To have a further syntactic similarity between the snippets, we cluster them via a TF-IDF scheme. Thus, for each cluster, we have text fragments similar to each other, single-authored and generic. We select five snippets from each of these clusters, one as a source and four as targets which we show to the human judges. We devise our experiments with these selections the following way:

- In Experiment 1, the source and only one of the target snippets are written by the same author.
- In Experiment 2, the source and only one target snippet are written from the same author, but also only another target snippet is published in the same journal as the source.

- In Experiment 3, all the snippets are written by different authors.

With Experiment 1, we want to identify whether human judges can recognize the snippet creator just from the style. Experiment 2, investigates whether papers written in the same journal have any impact on the writing style. With Experiment 3, we want to capture the features used by humans during their judgment. As a further filter, we remove the experiments for which there are trigger words that can bias human judges. 56 different annotators from 29 different countries participated in our evaluation performed in the crowdsourcing platform CrowdFlower[6]. Results showed a Krippendorff's alpha inter-rater agreement of 0.299 between the annotators, suggesting this outcome could have been generated from a random distribution. We tested such a possibility via a permutation test, which indicated that with a confidence of 72% the human judges did not rank randomly. Even if the choice doesn't look random, the result suggests that it is very hard, even for humans, to recognize very similar text snippets in specific cases. To further research on this result, we conducted a second pilot study, considered qualitative. Here, we wanted to eliminate any bias from the crowd by participating in the experiments ourselves. To check whether we can capture some clues, we also included stylometric hints in half of them. Unexpectedly, we performed worse than random in the normal experiments and almost random in those with feature hints. This result helped me answering my seventh research question:

RQ7: Is it possible, even for humans, to identify the authors in short text with high content similarity?

The next section draws the lesson learned from this thesis. The latter chapters contain the scientific dissemination that I have conducted and introduced in this thesis.

1.4 Conclusion and Future Work

In this thesis, I showed in various tasks the usage of three different features: semantic, syntactic, and stylistic. More precisely, in Chapter 2, I showed the

[6]Recently changed the name to "Figure Eight"

usage of OpenIE with the aid of semantic features in a sentiment analysis scenario. The presented system shows very high accuracy and performs similarly to the state of the art approaches for different categories. The results of this research were published in the Cognitive Computation Journal (Impact Factor 4.287, Q1) and Information Processing and Management Journal (Impact Factor 3.892, Q1) and assessed the usefulness of OpenIE in sentiment analysis. This work answers the first research question of this thesis:

RQ1: Can a combination of Open Information Extraction and semantic features achieve state-of-the-art results on ”aspect extraction” and ”polarity estimation” for generic opinion streams?

In Chapter 3, I present another research based on OpenIE. In this case, propositions are used for a summarization purpose. Here, I proposed a 4W summarization method that displays to human operators a list of ”event” blocks that answer the questions: ”Who?”, ”What?”, ”When?” and ”Where?”. The results of such a paper suggest that OpenIE provides promising results for the summarization tasks. This study was published in the European Conference on Knowledge Management (B-ranked). The outcome of this work, helps me answer my second research question:

RQ2: How do we adapt Open Information Extraction for generalized summarization in news (social) media monitoring?

Both studies presented above provide a good base for assessing the usefulness of propositions (extracted via OpenIE) not only in Information Retrieval but also in sentiment analysis and summarization. One of the learned lessons is that OpenIE has a very natural way to adapt to a sentiment analysis task. For summarization tasks, OpenIE cannot perform without additional preprocessing steps such as Temporal and Spacial annotations.

In Chapters 4 and 5, I use word embeddings, namely syntactic features. In Chapter 4, I show the use of word embeddings in a sentiment analysis scenario. Here, we have employed different traditional classifiers on a ”bag of words” like model with embedding features. Furthermore, we show the difference between using only words close to the aspect phrase and the whole text. This latter part of the study checks the effect in the polarity

of distant words from the aspect phrase. This paper was published in the EMSASW workshop in the European Semantic Web Conference (A-ranked), and helped me answer my third research question:

RQ3: To what extent does a "bag of words" approach using word embeddings predict the sentiment in tweets, and what is the difference with just using a window of words around the aspect phrase?

Chapter 5 shows the application of contextual word embeddings in a scenario with short text instances and small datasets. In this study, we confirm that BERT embeddings improve the baseline of this scenario. We further show that data augmentation has no effect or detrimental effect in this case. This study was published in the ACM Conference on Digital Libraries (A*-ranked), and answer my third and fourth research questions:

RQ4: Is it possible to design a neural-based architecture that can build effective models from small datasets that outperform state-of-the-art data augmented techniques?

RQ5: Would the use of data-augmentation techniques in contextual word embeddings for text classification have a detrimental effect on the overall effectiveness or, at least, to have no significant improvements?

Chapters 6, 7, and 8 analyze the writing style. In the first preliminary study (Chapter 6) published in the CLBib Workshop (at Internation Conference On Scientometrics and Infometrics), we visualize the writing style variation in scientific papers written by a different number of authors. This study hinted that there is a distinct change in the writing style if there are more authors in the article. This hint led to the work in Chapter 7, disseminated in the BIR workshop in the European Conference in Information Retrieval (B-ranked), where just by using stylistic features, we proved that we could identify the number of authors of a scientific paper 3 out of 4 times. These results answer my sixth research question:

RQ6: To which extent can we identify the number of authors of scientific papers just from their writing style?

Finally, Chapter 8 shows my conclusive research for this thesis, disseminated in the Scientometrics Journal (Impact Factor 2.77, Q1). This study addresses

the case of high content similarity between text snippets written by different authors. It concludes that even humans cannot identify the authors in such cases. This study answers my last research question addressed here:

RQ7: Is it possible, even for humans, to identify the authors in short text with high content similarity?

2 ReUS: a Real-time Unsupervised System For Monitoring Opinion Streams

Mauro Dragoni,
Marco Federici,
Andi Rexha

Abstract

One of the most important opinion mining research directions fall in the area of extraction of polarities referring to specific entities (aspects) contained in the analyzed texts. The detection of such aspects may be very critical, especially when documents come from unknown domains. Indeed, while it is possible to train domain-specific models for improving the effectiveness of aspects extraction algorithms in some contexts, in others, the most suitable solution is to apply unsupervised techniques by making such algorithms domain-independent. Recently, an emerging need is to exploit the results of aspect-based analysis in real-time environments. This led to the necessity of providing solutions supporting both an effective analysis of user-generated content and an efficient and intuitive way of visualizing collected data. In this work, we implemented an opinion monitoring service implementing (i) a set of unsupervised strategies for aspect-based opinion mining together with (ii) a monitoring tool supporting users in visualizing analyzed data. The aspect extraction strategies are based on the use of an open information extraction strategy. The effectiveness of the platform has been tested on

benchmarks provided by the SemEval campaign and have been compared with the results obtained by domain-adapted techniques.

2.1 Introduction

Online services like booking platforms, shops, and social media are becoming widely used by an increasing percentage of population. Each user of *the Internet* can express own opinions regarding products, services, or even other people thoughts. Opinions expressed by masses can lead other consumers to different choices, give direct feedback to the producers, or underline problematics of a service. For all these reasons, in the last few years, a lot of effort has been invested in understanding and extracting valuable data from user's reviews.

Opinion Mining and Sentiment Analysis are Natural Language Processing (NLP) tasks that aims to extract opinions from texts and to classify them with a value representing the overall polarity (*positive*, *negative*, or *neutral*) associated with a given subject (Pang, Lee, and Vaithyanathan, 2002; Erik Cambria, 2016). This research field attracted a lot of interest due to the possibility of applying developed strategies to a wide set of applications in different domains like marketing, politics, and social sciences. In the beginning, built applications aimed to compute overall polarity values then associated with a document. This strategy cannot distinguish the subject of each opinion and how users judged such a subject. This issue led to focus on the extraction of all subjects, namely *aspects*, from texts in order to equip developed systems with the possibility of computing aspects' polarities independently Hu and B. Liu, 2004.

Let us consider the following example:

Last weekend, I tried a new restaurant in downtown.
The place was awesome, but the quality of the food was quite poor.

The proposed example contains three aspects, *restaurant*, *place*, and *food*, and each aspect is associated with a specific opinion:

- *place* → *awesome*
- *food* → *quite poor*
- *restaurant* → no opinions. In this case, the polarity can be computed by averaging the polarities associated with the other aspects contained in the document.

For obtaining the opinion-based structure of the sentence, it is necessary to address two tasks: (i) the detection of the aspects, and (ii) the computation of the associated polarities. While the latter is easily supported by using opinion-based dictionaries (Section 2.3), the former requires different strategies. Many approaches presented in the literature, and discussed in Section 2.2, proposed supervised models for extracting aspects from text. Unfortunately, the use of a supervised approach clashes with real-world requirements. Firstly, the creation of a model requires annotated datasets containing aspects annotations for all possible domains. Nowadays, these datasets are not available except for a limited number of domains. Secondly, a document can have sentences belonging to many of these domains. Hence, the use of a single model is not feasible.

In light of these challenges, the development of approaches able to provide effective aspect extraction, polarity computation, and data visualization procedures is of interest for contexts where it is necessary to provide dashboards showing a real-time summary of opinion-based data-streams containing documents belonging to domains unknown a-priori. The use of an open information extraction strategy can be suitable for a real-time scenario. Indeed, here the system has to extract and to analyze information coming from all possible domains in an efficient way.

This work focuses on the creation of an opinion-based support system built upon the following three pillars.

- the design and the development of an open information extraction approach for supporting the detection of aspects within texts;
- the design and the development of a scalable platform able to process a high volume of opinion-based documents in real-time; and,
- the development of a data visualization interface supporting an easy access to the processed data.

The innovative aspect of this solution focuses on the combination of these three pillars in order to position this work as a state of the art platform for the real-time management of complex opinion-based documents.

One of the aim of the proposed system is to support different kind of users (managers, buyers, customers, etc.) with a multi-facet analysis of products' features. Indeed, the main issue when a product is judged with a single metric (e.g. overall document polarity) is that it does not allow users to obtain results tailored to their specific needs. For example, customers of an online shop could be interested in the *battery life* of a laptop rather than its overall quality. Currently, they do not have the possibility of obtaining this kind of information directly from the reviews, if users do not have the possibility of rating the specific *battery life* aspect.

The paper is structured as follows. In Section 2.2, we provide an overview of the opinion mining field with a focus on aspects extraction approaches. Section 2.3 introduces the background knowledge bases integrated into the proposed platform. Section 2.4 provides an overview of platform's components, while in Sections 2.5 and 2.6 we describe the strategy used for extracting aspects and the client application we developed for supporting users in monitoring the real-time data stream, respectively. Section 2.7 discusses the overall performance of our platform. Section 2.8 concludes the paper.

2.2 Related Work

In this section, we briefly review the main contributions in the field of sentiment analysis and opinion mining, firstly from a general standpoint and then with a particular attention to the social media scenario. A brief overview of significant recent contributions in the open information extraction field is also provided.

2.2.1 Sentiment Analysis and Opinion Mining

The topic of sentiment analysis has extensively been studied in the literature (B. Liu and L. Zhang, 2012; Erik Cambria, 2016), where several techniques have been proposed and validated.

Machine learning techniques are the most common approaches used for addressing the sentiment analysis problem. For instance, in Pang, Lee, and Vaithyanathan, 2002 and Pang and Lee, 2004 the authors compared the performance of Naive-Bayes, Maximum Entropy, and Support Vector Machines classifiers in sentiment analysis, using different features like considering only unigrams, bigrams, combination of both, incorporating parts of speech and position information, or considering only adjectives.

The recent massive growth of online product reviews paved the way for using sentiment analysis techniques in marketing activities. The issue of detecting the different opinions concerning the same product expressed in the same review emerged as a challenging problem. This task has been carried out by identifying the aspect of the product that a sentence in the opinion may refer to. In the literature, many approaches have been proposed: conditional random fields (CRF) (Jakob and Gurevych, 2010), hidden Markov models (HMM) (Jin, Ho, and Srihari, 2009), sequential rule mining (B. Liu, Hu, and Cheng, 2005), dependency tree kernels (Y. Wu et al., 2009), and clustering (Su et al., 2008).

Recently, the application of sentiment analysis approaches attracted a lot of interest also in the social networks research field (Go, Bhayani, and L. Huang, 2009). The use of social networks for expressing opinions and comments about products, political or social events, significantly increased in the last years. However, the analysis of the social network environment brought to light new challenges mainly related to (i) the different ways people express their opinions (i.e. *multi-modality*) and to (ii) the management of noisy data contained in social network texts (Barbosa and Feng, 2010).

The social dimension of the Web fostered the development of multi-disciplinary approaches combining computer and social sciences to improve the interpretation, recognition, and processing of opinions and sentiments expressed in social networks. The synergy between these approaches has been called

sentic computing (Erik Cambria and Amir Hussain, 2015). Sentic computing has been employed for addressing several cognitive-inspired problems like the classification of natural language text (Q. F. Wang et al., 2013) and the extraction of emotions from images (E. Cambria and A. Hussain, 2012).

Real-world solutions have been also developed. For example the authors of SENTILO (Gangemi, Presutti, and Recupero, 2014; Recupero et al., 2015) presented a semantic-based solution for extracting opinion frames from texts.

Different level of granularities have been considered: while some approaches operate at document level (M. Dragoni, 2015; Petrucci and Mauro Dragoni, 2015), other focus their goal on opinion classification by the means of a fine-grained analysis of the text at sentence level (Riloff, Patwardhan, and Wiebe, 2006; Wilson, Wiebe, and Hwa, 2006). Other approaches propose the use of fuzzy logic (Mauro Dragoni, A. G. Tettamanzi, and Costa Pereira, 2015; Mauro Dragoni, A. G. B. Tettamanzi, and Costa Pereira, 2014) or other aggregation techniques (Costa Pereira, Mauro Dragoni, and Pasi, 2009) to compute the score of each single word. In the case of sentence-level opinion classification, two different sub-tasks have to be addressed. The first one, called *subjectivity classification*, consists in detecting if the sentence is subjective or objective, while the second one focuses on determining if the expressed opinion is positive, negative, or neutral. *Subjectivity classification* rose great interest in the community (Riloff, Patwardhan, and Wiebe, 2006; Wilson, Wiebe, and Hwa, 2006). Systems implementing the capabilities of identifying opinion's holder, target, and polarity have been discussed (Aprosio et al., 2015).

Recent work on text modality used Convolutional Neural Network (CNN) (Chaturvedi, Erik Cambria, and Vilares, 2016) for sentiment related tasks such as sarcasm detection (Poria, Erik Cambria, Hazarika, et al., 2016) and aspect-based opinion mining (Poria, Erik Cambria, and Alexander F. Gelbukh, 2016). Several deep learning based approaches have been evaluated in Sentiment Analysis tasks. In Socher et al., 2013, Recursive Neural Networks are used to handle the syntactic tree structure of a sentence: following the generated parse tree, the different distributed representations of sentence parts are recursively built. The model is trained on the Stanford Sentiment Treebank, which has annotations on the whole parse tree. T. Chen et al., 2016

learn a distributed representation of reviews through Convolutional Neural Networks which are subsequently feed into Recurrent Neural Networks to learn distributed representation of the viewed products and of the opinion holders. In Poria, Erik Cambria, and Alexander F. Gelbukh, 2016 a 7-layer deep Convolutional Neural Network has been trained to identify the target of an opinion within a text fragment, in conjunction with some linguistic patterns. An Extreme Learning Machine approach implemented over the data analytics framework Apache Spark[1] has been proposed in Oneto et al., 2016. The approach deals with large amount of natural language text coming from the Social Web.

2.2.2 Opinion Mining in Social Media

The application of opinion mining approaches in social media became attractive by opening up new challenges due to the different ways people express their opinions (Barbosa and Feng, 2010). People use social networks to express their moods and opinion about recently purchased items or new products available on the marketplaces.

One of the first studies on opinion mining on micro-blogging websites has been discussed in Go, Bhayani, and L. Huang, 2009, where the authors presented a distant supervision-based approach for opinion classification on Twitter. In Thelwall et al., 2010 the authors presented SentiStrength. The described algorithm focuses on the detection of emotion strength in a social context. SentiStrength implements a machine learning approach aiming at optimizing opinion words weightings used for inferring the polarity of each message. Moreover, the approach implements a spelling correction method used to address the misspelling issue which often occurs on user-generated content.

However, the research in social media analytics has only recently started to employ aspect-based sentiment analysis in the process of attracting users. In particular, aspects extracted from opinions are exploited to attract users to follow the links related to products which have been judged interesting by users communities. A first attempt to exploit extracted aspects for better

[1] https://spark.apache.org/

orienting advertisements content is discussed in Fan and Chang, 2010. While in Sklar and Concepcion, 2014, the authors focused on tips instead of reviews. Their objective was recommending the right tips to the right people via the Foursquare platform, by taking into consideration the timeliness of user-provided tips and the users' tastes and social connections.

The increasing number of online product reviews enhanced the development of new opinion mining techniques due to their value in marketing activities. The detection of opinions regarding a specific product emerged as a real challenge. In this context, aspect extraction approaches achieved interesting results. The aspect extraction literature is divided into two distinct paths: supervised and unsupervised methodologies. The first one requires manually annotated data and it is mainly based on Conditional Random Fields (Jakob and Gurevych, 2010; Choi and Cardie, 2010; M. Zhang, Y. Zhang, and Vo, 2015; Mitchell et al., 2013), while the latter is focused on topic modeling (Mei et al., 2007; Titov and McDonald, 2008; F. Li, M. Huang, and Zhu, 2010; Mukherjee and B. Liu, 2012) and dependency relations (Y. Wu et al., 2009). Other approaches propose hidden Markov models (HMM) (Jin and Ho, 2009; Jin, Ho, and Srihari, 2009), sequential rule mining (B. Liu, Hu, and Cheng, 2005), dependency tree kernels (Y. Wu et al., 2009), clustering (Su et al., 2008), and genetic algorithms (Mauro Dragoni, Azzini, and A. Tettamanzi, 2010). With respect to these works, our approach relies on a scalable and unsupervised technique for detecting domain-specific aspects from opinion documents. This way, we are able to cope the challenge of deploying a light system into real-world general purpose scenarios.

In this paper, we bridge the aspect-based opinion mining and the user engagement areas by providing a smart way to exploit the knowledge extracted from user reviews. This work has been implemented and validated in a specific context. However, the approach described in Section 2.5 can be easily deployed in different scenarios.

2.2.3 Open Information Extraction

In the past years, a lot of research has been dedicated to constantly improve the performance of Open Information Extraction (OpenIE) systems. In the

beginning, shallow syntactical features such as part-of-speech tags were employed: *TextRunner* (Yates et al., 2007), WOE^{pos} (F. Wu and Weld, 2010), and *ReVerb* (Fader, Soderland, and Etzioni, 2011) making these systems highly efficient but poor in quality.

To improve the extraction quality, complex features, like dependency tree information, started to be exploited: *Kraken* (Akbik and Löser, 2012), *Ollie* (Mausam et al., 2012), *ClausIE* (Del Corro and Gemulla, 2013), and *CSD-IE* (Bast and Haussmann, 2013).

So far, the majority of the research focused on the English language, but other languages such as Spanish (Zhila and Alexander F Gelbukh, 2014), Chinese (M. Wang, L. Li, and F. Huang, 2014), and German (Falke et al., 2016) recently attract interest from the research community. The word presented in Gamallo, Garcia, and Fern'andez-Lanza, 2012 showed that OpenIE based on dependency trees is suitable for various languages besides English. They used a multilingual parser with a common output tag-set for the supported languages (English and Romance).

The multilingual OpenIE system *ArgOE* (Gamallo and Garcia, 2015) tries to be more open for different dependency parsers by using the CoNLL-X format. It manages to extract tuples in several languages with the same rule set, relying on a dependency parser that uses a common tag-set for five European languages. In Zhila and Alexander F Gelbukh, 2014 the Spanish system *ExtrHech* has been described. It works with part-of-speech-tagged input and semantic constraints, demonstrating that this approach achieves similar results for Spanish and English as well.

SCOERE (M. Wang, L. Li, and F. Huang, 2014) is an OpenIE system for the Chinese language. It uses a semi-supervised approach and focused on a fixed set of entities, namely person, organization, location and time. Falke et al., 2016 introduced *PropDE*, an OpenIE system for the German language. The *PropDE* system transfers the available set of extraction rules (*PropS* from Stanovsky et al., 2016) from English to German.

2.3 Material

Before presenting the system architecture and the approach designed for the specific aspect extraction task, we introduce here the resources we used for supporting the whole text analysis activity. We exploited four different resources: a stopwords list,[2] sentiment lexicons, a linguistic knowledge base, and a general-purpose natural language processing library.

2.3.1 Sentiment Lexicons

Sentiment Lexicons are used for associating each term with a polarity value. Terms having such an association are called *opinion words* and they are used for estimating the polarity of a given sentence. Associating a polarity value to a specific word is a task that has been addressed by different perspectives. The results have been the availability of different resources that can be easily integrated within real-world systems. In our platform, we decided to aggregate polarity values coming from three resources freely available: SenticNet (Erik Cambria, Poria, et al., 2016), the General Inquirer vocabulary [3] P.J, Dunphy, and Marshall, 1966, and the MPQA dictionary[4] (Deng and Wiebe, 2015).

SenticNet is a publicly available resource for opinion mining that exploits both artificial intelligence and semantic Web techniques to infer the polarities associated with common-sense concepts and to represent them in a semantic-aware format. The development of SenticNet was inspired by SentiWordNet (Baccianella, Esuli, and Sebastiani, 2010), a lexical resource in which each WordNet synset is associated to three numerical scores describing how objective, positive, and negative the terms contained in each synset are. The differences between SenticNet and SentiWordNet are basically three: (i) in SentiWordNet, each synset is associated to a three-valued representation (the objectivity of the synset, its positiveness, and its negativeness), while in SenticNet there is only one value belonging to the $[-1, 1]$

[2] The used stopwords list is available at http://www.lextek.com/manuals/onix/stopwords1.html
[3] http://www.wjh.harvard.edu/ inquirer/spreadsheet_guide.htm
[4] http://mpqa.cs.pitt.edu/corpora/mpqa_corpus/

interval for representing the polarity of the concept; (ii) SenticNet provides the sentiment model of more complex common-sense concepts, while SentiWordNet is focused on assigning polarities to WordNet synsets: for instance, in SenticNet, complex concepts like *make good impression*, *look attractive*, *show appreciation* , *being fired*, *leave behind*, or *lose control* are used for defining positive or negative situations; and (iii) completely neutral concepts are not reported. SenticNet contains almost 40,000 polarity concepts and it may be connected with any kind of opinion mining application. For example, after the de-construction of the text into concepts through a semantic parser, SenticNet can be used to associate polarity values to these and, hence, to infer the overall polarity of a clause, sentence, paragraph, or document by averaging such values.

The *General Inquirer* is an English-language dictionary containing almost 12,000 elements associated with their polarities in different contexts. Such dictionary is the result of the integration between the *Harvard* and the *Lasswell* general-purpose dictionaries as well as a dictionary of categories defined by the dictionary creators. When necessary, for ambiguous words, specific polarity for each sense is specified. For every word, a set of tags is provided in the dictionary. Only a subset of them are relevant to the opinion mining topic and, thus, exploited in this work:

- Valence categories: the two well-known *positive* and *negative* classifications.
- Semantic dimensions: these tags reflect semantic differential findings regarding basic language universals. These dimensions are: *hostile*, *strong*, *power*, *weak*, *submit*, *active*, and *passive*. A word may be tagged with more than one dimension, if appropriate.
- Words of pleasure: these tags are usually also classified positive or negative, with virtue indicating strength and vice indicating weakness. They provide more focus than the categories in the previous two bullets. Such categories are *pleasure*, *pain*, *feel*, *arousal*, *emotion*, *virtue*, *vice*.
- Words reflecting presence or lack of emotional expressiveness: these tags indicate the presence of overstatement and understatement; trivially, such tags are *overstated* and *understated*.

Other categories indicating ascriptive social tags rather than references to

places have been considered out of the scope of the opinion mining topic and have not been considered in the implementation of the approach.

Finally, *MPQA* is a sentiment lexicon built for Multi-Perspective Question Answering purposes. The lexicon contains around 8,222 terms annotated with their polarity (*positive*, *negative*, and *neutral*) and with their intensity level (*strong* and *weak*) and a set of 10,000 sentences manually annotated through the proposed annotation scheme. Indeed, besides the classic association $\langle word, polarity \rangle$, the MPQA lexicon implements a detailed annotation scheme that identifies key components and properties of opinions, emotions, sentiments, speculations, evaluations, and private states Quirk et al., 1985. This annotation scheme covers a broad and useful subset of the range of linguistic expressions and phenomena employed in naturally occurring text to express opinion and emotion. The proposed annotation scheme is relatively fine-grained, annotating text at the word- and phrase-level rather than at the level of the document or sentence. For every expression of a private state in each sentence, a private state frame is defined. A private state frame includes the source of the private state (i.e., that whose private state is being expressed), the target (i.e., what the private state is about), and various properties involving intensity, significance, and type of attitude. An important property of sources in the annotation scheme is that they are nested, reflecting the fact that private states and speech events are often embedded in one another. The representation scheme also includes frames representing material that is attributed to a source, but is presented objectively, without evaluation, speculation, or other type of private state by that source.

The lists of terms contained in the resources presented above do not overlap completely. The strategy implemented within our platform considers words with a non-zero polarity value in at least one of the integrated resources. For example, the word *third* is not present neither in MPQA nor in SenticNet and has a polarity of 0 according to the General Inquirer. Consequently, it is not a valid opinion word. On the other hand, the word *huge* has a positive value of 0.069 in SenticNet, a negative value of -1 in MPQA and a value of 0 in the General Inquirer, therefore, it is evaluated as opinion word even if lexicons express contrasting values. SenticNet already implements a continuous representation of polarity values. MPQA uses a discrete scale $[-1, 0, 1]$ that has been extended to $[-1, -0.5, 0, 0.5, 1]$ by halving -1 and 1

when the *weak* intensity level is present. For the General Inquirer the same strategy adopted for the MPQA lexicon has been adopted by exploiting the semantic dimension of the dictionary for halving the -1 and 1 values. Finally, the three values are aggregated by using the mean.

2.3.2 WordNet

WordNet[5] (Fellbaum, 1998) is a large lexical database of English nouns, verbs, adjectives, and adverbs grouped into sets of cognitive synonyms called *synsets*, where each synset expresses a distinct concept. In particular, each synset represents a list of synonyms, intended as words that denote the same concept and that are interchangeable in many contexts. WordNet contains around 117,000 synsets linked to each other by a small set of *conceptual relations*, i.e., synonymy, hypernymy, hyponymy, etc.. Additionally, a synset contains a brief definition (*gloss*) and, in most cases, one or more short sentences illustrating the use of the synset members. Words having several distinct meanings are represented in as many distinct synsets. Even if WordNet superficially resembles a thesaurus, there are some important distinctions with respect to it. Firstly, WordNet does not define links between words, but between specific senses of words; this way, words that are found in close proximity to one another in the network are semantically disambiguated. Secondly, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than the similarity of their meanings. In the implemented system, Wordnet's compound names list has been used to detect word sequences that represent a single concept.

2.3.3 Stanford Core NLP

The preliminary textual analysis, consisting in converting the raw input text in an annotated and structured representation, is performed through the Stanford Core Natural Language Processing Library (Christopher D.

[5] https://wordnet.princeton.edu/

Manning et al., 2014). Stanford CoreNLP is an integrated framework providing a wide range of natural language analysis tools. Each functionality is provided by a specific module. Below, we show the four modules of the CoreNLP library adopted within our system.

The *Pos Tagger* (Part Of Speech Tagger) is a software module aiming to assign a part of speech tag (such as noun, verb, adjective, etc.) to every word of a given sentence (Kristina Toutanova and Singer, 2003). The *Coref Annotator* (Co-reference resolution Annotator) generates co-reference Chain Annotations representing groups of words referring to the same entity (Clark and Christopher D. Manning, 2015). Chains are used to resolve pronoun references. The *Parse Annotator* (Parser Annotator) (Klein and Manning., 2003) provides full syntactic analysis generating a tree grammar dependencies structure. Finally, the *Depparse Annotator* (Dependency Parser Annotator) (D. Chen and Christopher D Manning, 2014) provides a representation of grammatical relations between words in a sentence producing graphs like the one shown in Figure 2.4.

2.4 System Architecture

The system presented in this work implements a set of modules for supporting the gathering, the processing, and the analysis of opinion-based document streams. In particular, we focused on the Amazon website. Figure 2.1 shows an abstract overview of these modules. Reviews collected in real-time from the Amazon website are given as input to the ***Data Manager Module*** that is responsible of parsing raw documents and of enriching them with further metadata. Processed data are saved into a knowledge repository in order to make them available to a ***Web Service*** that is responsible of exposing the structured knowledge as result of client requests.

The workflow works in the following way. The stream of reviews are given as input to the *Data Manager Module*. This module is composed by two components: the Document Analyzer Pipeline and the Document Enricher. This former is responsible of applying the open information extraction strategy, together with other natural language processing tools, for extracting

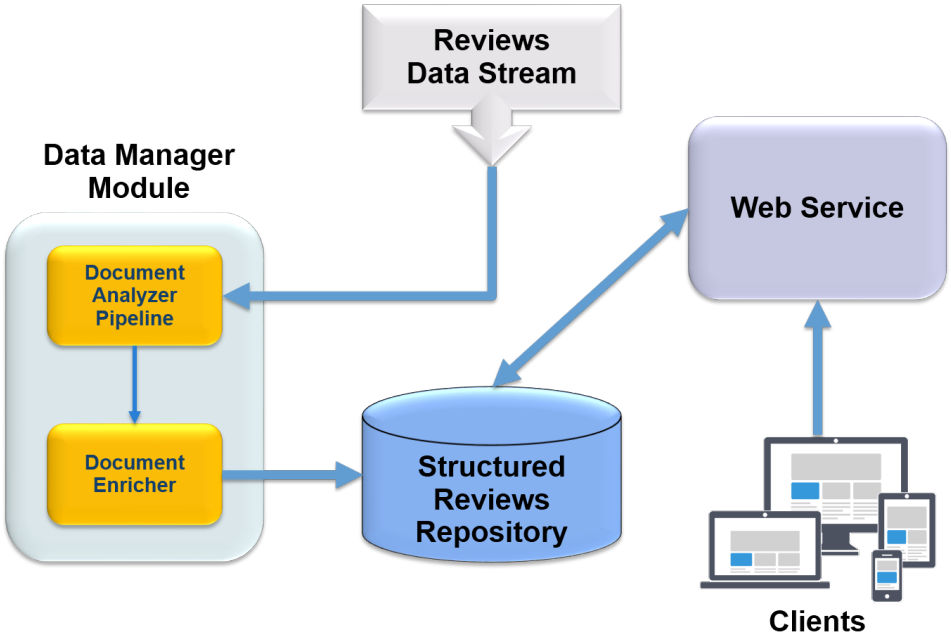


Figure 2.1: Overview of the implemented platform.

tuples containing the aspects mentioned in the text and the associated polarities. Details about this module are provided in Section 2.5.

Once a review is analyzed, the result is sent to the Document Enricher component that is responsible of linking information extracted from text to the product for which the review has been provided. The linking operation consists of retrieving the product name, the domain, the category, and the review score. This operation has been implemented on top of the Amazon Product API. Here, given the product's id contained in the review's metadata, it is possible to retrieve the product's information mentioned above.

The output of the *Data Manager Module* is the structured representation shown in Figure 2.2. Each object is then stored into the repository.

Leaves of the tree contain the label of the opinion word and its polarity. These are associated with the respective aspects contained within the connected upper-layer. All aspects are finally associated with the product entity.

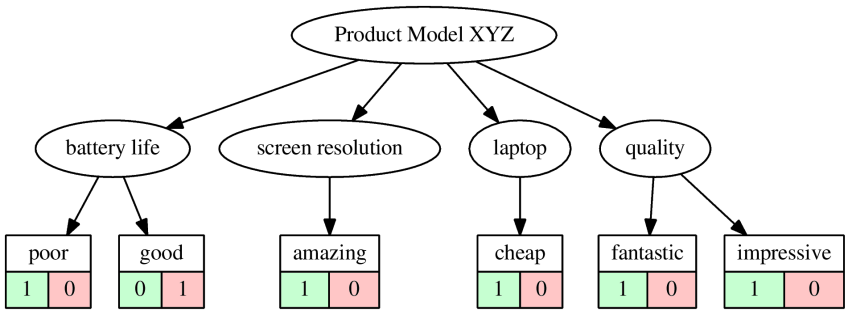


Figure 2.2: Example of the tree structure extracted by NLP Module

The content of the repository is then exploited by final users through the *Web Service* integrated into the platform. The *Web Service*, and the client application briefly presented in Section 2.6, enables users to access data and to have a real-time visualization about the opinion trends associated with products' categories, items, or specific features of items.

2.5 Document Analyzer Pipeline

Here, we present the approach implemented for extracting aspects from document content. The overall aspect extraction approach relies on the NLP pipeline shown in the middle layer of Figure 2.3.

As it is shown on the bottom layer, the implemented pipeline exploits the three linguistic resources introduced in Section 2.3. These resources are used by the Stanford Core NLP Library Christopher D. Manning et al., 2014 shown in the top layer of Figure 2.3.

The pipeline is composed by the following five phases:

- Aspect Extraction. This first step is the most important one and it consists on detecting the correct aspects contained in the text and the associated opinion words. Details about this step are provided below where we present the open information extraction algorithm adopted for analyzing provided text.

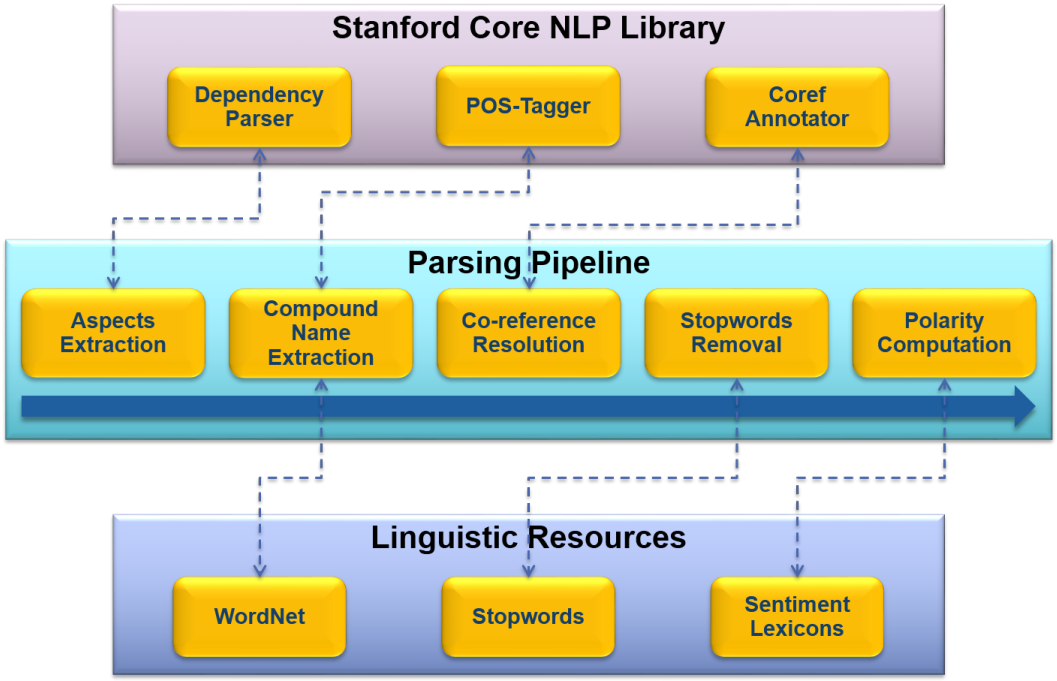


Figure 2.3: The NLP pipeline implemented within the proposed platform aiming to extract aspects and compute their polarity from the analyzed textual resources.

- Compound Noun Extraction. The second step consists in detecting the presence of compound names. This step is supported by the use of the POS-Tagger module provided by the Stanford Core NLP library and by WordNet (both introduced in Section 2.3). When two consecutive words are tagged as nouns by the POS-Tagger, their composition is searched within the WordNet dictionary. If the compound expression is found, it is tagged as compound name and used as a unique token, otherwise not.
- Co-reference Resolution. This step consists in associating pronouns with the related noun (or compound noun). This is necessary for detecting all associations between opinion words and aspects. This operation is completely supported by the Coref Annotator. Refinements of the adopted algorithm are out of scope of this paper and they are part of future work.
- Stopwords Removal. Once compound names have been detected and

pronouns have been replaced with the right terms, the pipeline removes all stopwords from the text by exploiting the list mentioned above.

- Polarity Computation. Finally, this step is responsible for computing the polarity associated with each aspect extracted during the previous steps. The overall polarity of an aspect A is computed by aggregating the single polarities of the opinion words associated with A. Single polarity values are extracted and aggregated from the sentiment lexicons as described in Section 2.3.

2.5.1 The Open Aspect Extraction Strategy

The Open Aspect Extraction component uses a generic solution for identifying possible aspects in the user's opinion. This component implements an OpenIE strategy for supporting this task. OpenIE is a NLP branch of research that tries to determine meaningful patterns over parsing structure of a sentence and morphological characteristics.

The developed algorithm analyzes the structure of the grammar dependencies graph generated by the CoreNLP library for extracting the connections between aspects and opinions. Each dependency extracted by the CoreNLP library can be expressed by a triple: $\{Relation\_Type, Governor, Dependant\}$[6].

The generated dependency graph is then processed by applying a set of rules for determining if the content of each node is supposed to be an aspect, an opinion, or nothing. These rules can be considered as a representation of the most common patterns that can be used for detecting pairs of the type *aspect-opinion_word*. The choice of these three rules allows at the same time to have a system that is efficient in processing document content and effective in covering content structure. Indeed, results of an in-vitro experiment shown the by disabling one of the rules the effectiveness of the system dramatically decreases. Hence, given a dependency node n, the algorithm checks if one of the following rules subsists:

[6]The meaning of each element of the triple together with all the possible relation type, can be found in the official Stanford Document available at http://nlp.stanford.edu/software/dependencies_manual.pdf

Rule 1: If the relation type is an adjectival modifier ("amod"), a connection between an aspect and an opinion word persists if and only if the governor is an aspect and the dependant has a polarity value in at least one of the sentiment lexicons.

Rule 2: If the relation type is a nominal subject ("nsubj"), a connection between an aspect and an opinion word persists if and only if the governor has a polarity value in at least one of the sentiment lexicons and the dependant is an aspect.

Rule 3: If the relation type is a direct object ("dobj"), a connection between an aspect and an opinion word persists if and only if the governor has a polarity value in at least one of the sentiment lexicons and the dependant is an aspect.

Figure 2.4 shows the result obtained by applying these three rules to our running example and Figure 2.5 summarizes only the valid relationships extracted from the grammar dependencies graph. We reported with a dotted line also the relationship between *I* and *enjoyed*. Actually, this relationship is not valid because *I* is tagged as *personal pronoun* but within the sentence such a pronoun is not resolved.

The color code used in the figures is the following: light red nodes are nouns or noun phrases that have not been detected as *aspect* by the system; red nodes are nouns that have been detected as *aspect* by the system; green nodes are verbs for which a polarity value is present in the sentiment lexicons; and, blue nodes are adjectives for which a polarity value is present in the sentiment lexicons.

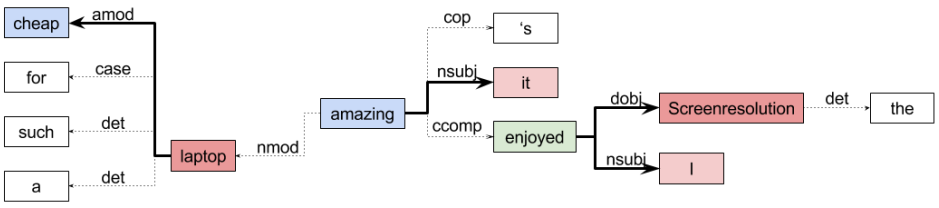


Figure 2.4: Dependency graph generated by the implemented approach.

Finally, the extracted relationships can be summarized as follows:

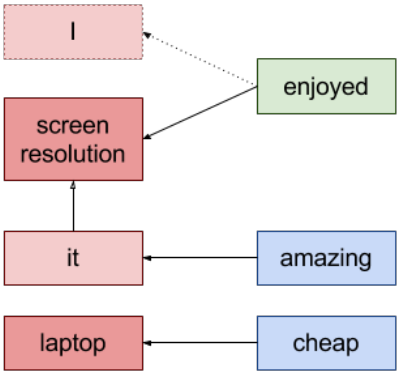


Figure 2.5: Relationships generated by the implemented approach.

$$\text{laptop} \leftarrow \{\text{cheap}\}$$
$$\text{screenresolution (it)} \leftarrow \{\text{amazing, enjoyed}\}$$

These associations allow our system to infer, for both aspects *laptop* and *screen_resolution*, a positive polarity.

2.6 Client User Interface

The platform has been equipped with a web-based client application for supporting users during the analysis of the processed data. Users can query the repository by means of a controlled query interface built on a single-page web application that provides all data visualization functionalities.

The client interface has been designed with the aim of being very simple and intuitive. First of all, users have to select a category from the related list. They can eventually specify an aspect of their interest and, then, submit their request. The web service will provide the list of products according to the specified category and the requested aspect. The client application is then in charge of organizing the response data as shown in Figure 2.6. The harmonization of the terms provided by the users (plural forms, synonyms, etc.) is performed by using the WordNet (Fellbaum, 1998) lexicalizer.

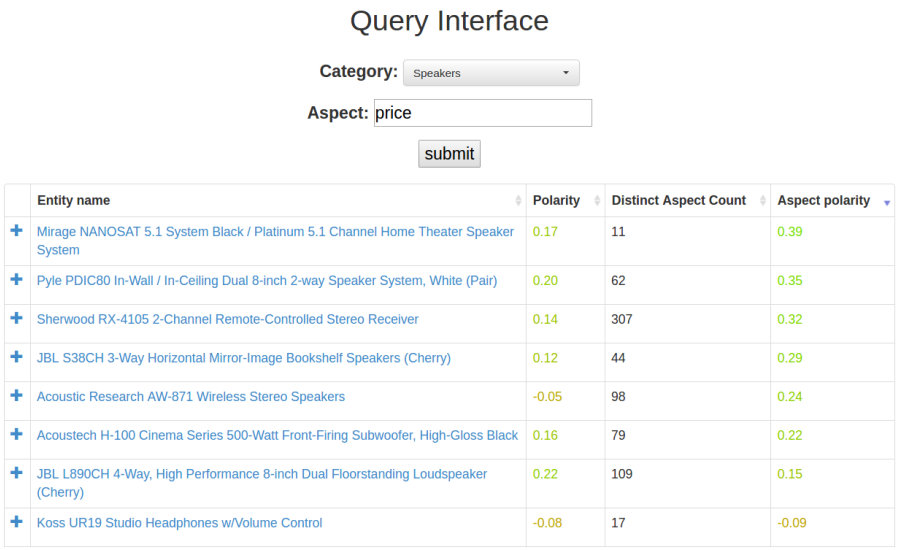


Figure 2.6: Example of query results.

Each row can be ordered according to the product name, the number of reviewed aspects, the average polarity, or the polarity of the aspect provided by the user, if any. These two last metrics are particularly useful because they represent, respectively, the customers' overall opinion of the product and their appreciation of the selected aspect.

A complete visualization of the product's opinion hierarchy is generated by clicking on its name. Figure 2.7 shows the editable tree-view obtained by selecting a specific product. Users can hide opinions for a better visualization of larger trees. Moreover, colors have been added to give an immediate feedback on polarity values.

For retrieving further product details, each row can be expanded for showing the details of every single aspect extracted from the reviewed entity. The aspects sub-table shows aspect's name, average polarity, and the number of related opinion.

Single opinions can be visualized by expanding the aspect's row as shown in Figure 2.8.

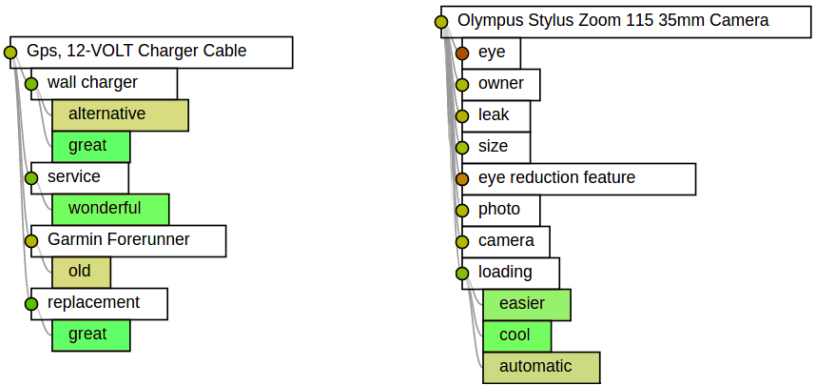
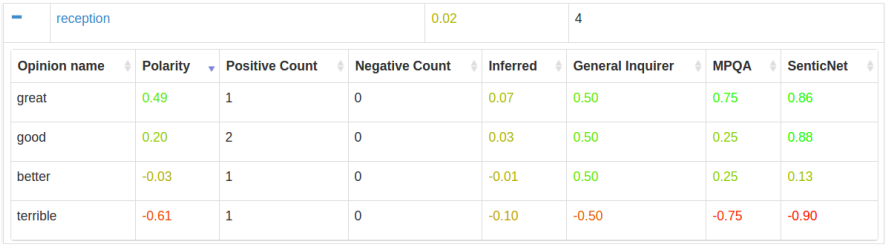


Figure 2.7: Example of the tree visualization of the data regarding two products



reception 0.02 4

Opinion name	Polarity	Positive Count	Negative Count	Inferred	General Inquirer	MPQA	SenticNet
great	0.49	1	0	0.07	0.50	0.75	0.86
good	0.20	2	0	0.03	0.50	0.25	0.88
better	-0.03	1	0	-0.01	0.50	0.25	0.13
terrible	-0.61	1	0	-0.10	-0.50	-0.75	-0.90

Figure 2.8: Example of the opinion subtable

Once displayed, opinion's table presents a slightly different structure. Each row reports opinion's name and its polarity as well as the values associated with the same word in the other lexicons resources. This way, it is possible to do an immediate comparison of such values. The count of positive and negative occurrences of the specific opinion are shown in two separates columns.

From the technological perspective, the main component of the web interface has been developed with Spark Micro Framework[7]. This Java library facilitate the creation of a simple REST service to manage client's requests.

[7]http://sparkjava.com/

Each call is binded to a specific query function which automatically maps the results to a JSON serializable object. The client side JavaScript code has been written following MVC pattern. AJAX requests and responses are handled by JQuery, Bootstrap[8], and Bootstrap-table[9] JavaScript libraries. These libraries are in charge of managing the presentation layer. Finally, the D3.js[10] library has been used to represent complex product data.

2.7 Evaluation

In this Section, we present the evaluation of the proposed system. Such a system is evaluated under different perspectives aiming to show the efficiency and effectiveness of the implemented modules:

- *Aspect extraction*. The OpenIE approach is in charge of detecting aspects within text. Such a task is important for defining, later in the analysis process, which aspects are the most significant ones and which are the opinion words associated with them. This evaluation task focused on measuring the effectiveness of the aspect-extraction approach.
- *Polarity detection*. The computation of the aspect's polarity enables the detection of which product features are strong or weak. The Sentiment Module is in charge of inferring the polarity of each aspect given the context in which such an aspect is included. Here, we measured the capability of our approach to infer the correct polarity.
- *Lessons Learned*. Besides the effectiveness of the technological components, we provide a discussion about the usability of the user interface and the direction we intend to follow for the evolution of the platform presented in this article. Here, the web-based tool has been evaluated by a group of 42 users of different expertises that answered to a survey. We report the most important feedback we collected.

The OpenIE module has been evaluated on two benchmarks: the SemEval

[8]http://getbootstrap.com/
[9]https://github.com/wenzhixin/bootstrap-table
[10]https://d3js.org/

2015 Task 12 [11] and SemEval 2016 Task 5 [12] datasets [13]. Both benchmarks required the detection of aspects from text belonging to the *Restaurant* and *Laptop* domains and the computation of the associated polarity. Then, concerning the SemEval 2015 Task 12 dataset, the polarity computation was requested also for the *Hotel* domain. In Section 2.7.1, we report the results obtained on the aspect detection task. For the SemEval 2015 Task 12 dataset the precision, recall, and f-measure metrics were available, while for the SemEval 2016 Task 5 dataset only the f-measure was reported in the official evaluation report. Then, in Section 2.7.2, we report the results obtained on the polarity computation task. Here, the system accuracy has been reported. Finally, besides the systems participated to the SemEval challenges, we included also a comparison with other two approaches available in the state of the art that are particularly relevant for our use case. Technical details about such approaches are presented in Hu and B. Liu, 2004 and Qiu et al., 2011.

We applied the paired t-test for measuring if the obtained results are significant. In each table, we used the following notation near the results obtained by the systems we compared: $--$ and $-$ means that the gap is significantly worse with a p-value of 0.01 or 0.05 respectively. While, $++$ and $+$ means that the gap is significantly better with a p-value lower than 0.01 or 0.05 respectively.

2.7.1 Evaluation on Aspect Extraction

Tables 2.1, 2.2, and 2.3 report the results obtained by our system on the SemEval benchmarks. As mentioned above, the algorithm has been tested on both the *Restaurant* and *Laptop* domains.

The overall performance are in line with the best systems participating in the evaluation campaigns and, in both cases, our approach obtained the best F-measure on the *Laptop* domain. It is also important to highlight that all the

[11] http://alt.qcri.org/semeval2015/task12/

[12] http://alt.qcri.org/semeval2016/task5/

[13] Concerning the evaluation on the SemEval 2016 Task 5 dataset, we applied our system to the Subtask 1, Slots 2 and 3 only. We worked in this way because the other tasks and slots aimed to verify system facilities that were out of scope of this paper.

System Acronym	Restaurant		
	Precision	**Recall**	**F-Measure**
IHS-RD-Belarus	**0.7095**	0.3845^{--}	0.4987^{--}
LT3 pred	0.5154^{--}	0.5600	0.5367^{--}
NLANGP	0.6385^{-}	$\mathbf{0.6154}^{+}$	**0.6268**
sentiue	0.6332^{-}	0.4722^{-}	0.5410^{--}
SIEL	0.6440^{-}	0.5135^{-}	0.5714^{-}
TJUdeM	0.4782^{--}	0.5806	0.5244^{--}
UFRGS	0.6555^{-}	0.4322^{--}	0.5209^{--}
UMDuluth	0.5697^{--}	0.5741^{+}	0.5719^{-}
V3	0.4244^{--}	0.4129^{--}	0.4185^{--}
Hu and B. Liu, 2004	0.5795^{--}	0.5287	0.5529^{-}
Qiu et al., 2011	0.6182^{-}	0.5329	0.5724^{-}
System Results	0.6895	0.5368	0.6036

Table 2.1: Results obtained on the aspect extraction task, for the *Restaurant* domain on the SemEval 2015 Task 12 benchmark. For each dataset, we reported Precision, Recall, and F-Measure. Acronyms refer to the systems participated in the SemEval 2015 Task 12 competition. Technical details about the participant systems can be found in the SemEval 2015 Proceedings (http://aclweb.org/anthology/S/S15/)

systems we compared to, apply a domain-specific supervised approaches for extracting aspects, while our approach implements an unsupervised technique. This peculiarity enables the possibility of implementing the system in any environment without the requirement of training a new model.

While on the *Laptop* domain, our system outperforms the others, a different scenario occurs for the *Restaurant* domain where our system loses around 3% and 6% on the two datasets, respectively. A more in-depth analysis of the results, in general, showed that the majority of the errors compared to other systems are caused by the extraction of aspects resulted as false-positive. This observation was not unexpected. Indeed, one of the most common issues in unsupervised aspect-based approaches resulted in the extraction of false-positive elements (Q. Liu et al., 2015). By analyzing the possible consequences of this weakness, we suppose that this may lead to poor effectiveness of components that exploit the aspect extraction module's

System Acronym	Laptop		
	Precision	**Recall**	**F-Measure**
IHS-RD-Belarus	0.5548^{--}	0.4483	0.4959^{-}
NLANGP	0.6425^{-}	0.4208	0.5086
sentiue	0.5773^{--}	0.4409	0.5000
TJUdeM	0.4489^{--}	$\mathbf{0.4820}^{+}$	0.4649^{-}
UFRGS	0.5066^{--}	0.4040	0.4495^{--}
V3	0.2710^{--}	0.2310^{--}	0.2494^{--}
Hu and B. Liu, 2004	0.6247^{-}	0.3589^{--}	0.4559^{--}
Qiu et al., 2011	0.6412^{-}	0.3773	0.4751^{--}
System Results	**0.6702**	0.4157	**0.5131**

Table 2.2: Results obtained on the aspect extraction task, for the *Laptop* domain on the SemEval 2015 Task 12 benchmark. For each dataset, we reported Precision, Recall, and F-Measure. Acronyms refer to the systems participated in the SemEval 2015 Task 12 competition. Technical details about the participant systems can be found in the SemEval 2016 Proceedings (http://aclweb.org/anthology/S/S15/)

outcomes.

Concerning the results of the paired t-test, in the cases where our system did not obtain the best result, we performed the test by taking as reference the best system. While, in the cases where our system obtained the best result, we performed the test by taking as reference the runner-up system. Concerning the results show in Table 2.1, the precision value has been compared with the *IHS-RD-Belarus* system by resulting not significant (p-value = 0.159), while the recall value has been compared with the *NLANGP* system and in this case the difference resulted significant (p-value = 0.020). Results presented in Table 2.2 were compared with the *NLANGP* system for the precision value and with the *TJUdeM* system for the recall value. Both differences resulted significant at the t-test with p-values of 0.043 and 0.016, respectively. The same happened for the results shown in Table 2.3. Here, the comparison has been performed only against the *NLANGP* system and for both domain the improvements resulted significant with p-values of 0.041 and 0.047. Overall, by considering the F-Measure values, our system obtained significant improvement on the *Laptop* domain, while for the *Restaurant* and *Hotel* domains both positive and negative differences with

System Acronym	**F-Measure** *Restaurant*	**F-Measure** *Laptop*
NLANGP	**0.7234**$^{+}$	0.5194^{-}
AUEB	0.7044^{+}	0.4911^{--}
UWB	0.6709	0.4790^{--}
GTI	0.6655	n.a
Senti	0.6655	n.a
bunji	0.6488	0.3959^{--}
DMIS	0.6350^{-}	n.a
XRCE	0.6198^{--}	n.a
UWate	0.5707^{--}	n.a
KnowC	0.5682^{--}	n.a
TGB	0.5505^{--}	n.a
BUAP	0.5025^{--}	0.2679^{--}
basel	0.4407^{--}	0.3748^{--}
IHS-R	0.4381^{--}	0.3902^{--}
IIT-T	0.4260^{--}	0.4391^{--}
SeemGo	0.3433^{--}	0.4150^{--}
SYSU	n.a	0.4907^{--}
BUTkn	n.a	0.4840^{--}
NileT	n.a	0.4720^{--}
INSIG	n.a	0.4586^{--}
LeeHu	n.a	0.4375^{--}
UFAL	n.a	0.2698^{--}
CENNL	n.a	0.2691^{--}
Hu and B. Liu, 2004	0.6321	0.5141^{-}
Qiu et al., 2011	0.6427	0.5187^{-}
System Results	0.6687	**0.5692**

Table 2.3: Results obtained on the aspect extraction task, for both the *Restaurant* and *Laptop* domains on the SemEval 2016 Task 5 benchmark. For each dataset, we reported the F-Measure. Acronyms refer to the systems participated in the SemEval 2016 Task 5 competition. Technical details about the participant systems can be found in the SemEval 2016 Proceedings (http://aclweb.org/anthology/S/S16/)

respect to the other systems did not result statistical significant. However, by considering the unsupervised nature of our approach, with respect to the compared systems that are all supervised, we may consider our

strategy feasible for being implemented in a real-world general purpose environment.

2.7.2 Evaluation on Polarity Computation

Tables 2.4 and 2.5 report the results obtained on the polarity computation task. The approach has been evaluated on the two benchmarks described in the preamble of this section. Here, we measured the accuracy of the polarity computation algorithm: given the set of opinion words associated with an aspect, the polarity is computed by aggregating the polarity values of each opinion words.

System Acronym	**Acc. *Restaurant***	**Acc. *Laptop***	**Acc. *Hotel***
ECNU	0.7810	0.7829^{-}	n.a
EliXa	0.7005^{--}	0.7291^{--}	0.7965^{--}
lsislif	0.7550	0.7787^{-}	**0.8584**
LT3	0.7502	0.7376^{--}	0.8053^{-}
sentiue	**0.7869**	0.7934^{-}	0.7876^{--}
SIEL	0.7124^{-}	n.a	n.a
SINAI	0.6071^{--}	0.6585^{--}	0.6372^{--}
TJUdeM	0.6887^{--}	0.7323^{--}	n.a
UFRGS	0.7171^{-}	0.6733^{--}	0.6578^{--}
UMDuluth	0.7112^{-}	n.a	0.7139^{--}
V3	0.6946^{-}	0.6838^{--}	0.7109^{--}
wnlp	0.7136^{-}	0.7207^{--}	0.5546^{--}
Hu and B. Liu, 2004	0.6936^{-}	0.7587^{-}	0.7896^{--}
Qiu et al., 2011	0.6997^{-}	0.7654^{--}	0.7947^{--}
System Results	0.7794	**0.8589**	0.8524

Table 2.4: Results obtained on the computation of polarities associated with single aspects on the SemEval 2015 Task 12 benchmark. For each dataset, we reported the accuracy obtained in computing polarities (*positive*, *negative*, or *neutral*). Acronyms refer to the systems participated in the SemEval 2015 Task 12 competition. Technical details about the participant systems can be found in the SemEval 2015 Proceedings (http://aclweb.org/anthology/S/S15/)

System Acronym	Acc. *Restaurant*	Acc. *Laptop*
XRCE	**0.8813**	n.a
IIT-T	0.8673	0.7840^{-}
NileT	0.8545	0.7740^{-}
IHS-R	0.8394	0.7790^{-}
ECNU	0.8359^{-}	0.7815^{-}
AUEB	0.8324^{-}	0.7690^{-}
INSIG	0.8207^{-}	0.7840^{-}
UWB	0.8184^{-}	0.7378^{-}
SeemGo	0.8114^{-}	0.7216^{--}
bunji	0.8102^{-}	0.7029^{--}
TGB	0.8091^{--}	n.a
UWate	0.8033^{--}	0.7129^{--}
DMIS	0.7998^{--}	n.a
Senti	0.7811^{--}	0.7428^{-}
LeeHu	0.7811^{--}	0.7591^{-}
basel	0.7648^{--}	0.7004^{--}
AKTSKI	0.7171^{--}	n.a
COMMIT	0.7055^{--}	0.6754^{--}
SNLP	0.6997^{--}	n.a
GTI	0.6997^{--}	0.6729^{--}
CENNL	0.6391^{--}	0.5993^{--}
BUAP	0.6089^{--}	0.6280^{--}
Hu and B. Liu, 2004	0.8318^{-}	0.7184^{--}
Qiu et al., 2011	0.8162^{-}	0.7458^{-}
System Results	0.8710	**0.8108**

Table 2.5: Results obtained on the computation of polarities associated with single aspects on the SemEval 2016 Task 5 benchmark. For each dataset, we reported the accuracy obtained in computing polarities (*positive*, *negative*, or *neutral*). Acronyms refer to the systems participated in the SemEval 2016 Task 5 competition. Technical details about the participant systems can be found in the SemEval 2016 Proceedings (http://aclweb.org/anthology/S/S16/)

Results demonstrated the effectiveness of the polarity computation strategy implemented into the proposed system. The system obtained the best performance on the *Laptop* domain in both benchmarks, while the gap with the

best systems on the other domains is always lower than 2%. After a detailed analysis of the results, we noticed that the reason for which our approach performs better on the *Laptop* dataset is due to the simple language used for describing product features. Indeed, in the *Restaurant* dataset opinions are expressed in a more articulated way and sometimes the approach fails to detect the right polarity. Part of future effort will be dedicated to improve our system in this direction.

We performed the same t-test described in the previous subsection also to the polarity computation results. Concerning the results reported in Table 2.4, we compared our system with *SENTIUE* on the *Restaurant* and *Laptop* domains, while for the *Hotel* domain the comparison has been done with the *LSISLIF* system. For both the *Restaurant* and *Hotel* domains, the results didn't differ significantly (p-values of 0.218 and 0.227, respectively), while for the *Laptop* domain the improvement is significant (p-value of 0.029). Finally, concerning the results shown in Table 2.5, we compared our system with *XRCE* for the *Restaurant* domain and with *INSIG* for the *Laptop* domain. In the first case the difference was not significant (p-value = 0.189), while in the second case the improvement obtained by our system was significant (p-value = 0.038). Overall, in almost all cases our approach significantly improved the other systems. In particular, on the *Laptop* domain in both cases all improvements are significant for a p-value of at least 0.05.

2.7.3 Lessons Learned

Early in this section, we demonstrated the suitability of the components integrated within the proposed platform. Besides such validation tasks, we interviewed a group of 42 users for collecting feedback about possible improvements on the client side. In particular, what we collected from users can be recognized in two main aspects: (i) efficient management of data streams, and (ii) understandability of the user interface.

Architecture Efficiency The scenario used in this first prototype focused on using document sets having a limited number of items. By switching from a test environment to a more complex one, we noticed that the time

needed for extracting all aspects increased significantly. This issue was related to the necessity of detecting, for each aspect that was already extracted, the presence of further opinions connected to him. While a possible solution might be the parallelization of this task, some tricks have to be appliedIndeed, the constraint of analyzing documents by keeping the timing order in which they have been generated requires to perform some checks based on the number of documents that we want to analyze at a certain time. Thus, by having, for example, a window of n documents that we want to parallelize, a possible strategy is to verify if there are conflicts between the aspects extracted from such documents. This way, we would be able to update aspect-based information without confusing the system. For completeness, we report data concerning the scalability of the system. We run the scalability test on a server equipped with a double Xeon X5650 and 32Gb of RAM and we measured the time necessary for processing the 1,000,000 documents contained within the DRANZIERA dataset (Mauro Dragoni, A. Tettamanzi, and Costa Pereira, 2016). We tested three systems: the one we propose, the approach presented in Hu and B. Liu, 2004, and the one presented in Qiu et al., 2011. Our system completed the processing operation in 23 minutes and 27 seconds, the algorithm of Hu and B. Liu, 2004 in 63 minutes and 45 seconds, and the algorithm of Qiu et al., 2011 in 92 minutes and 31 seconds. Thus, we may state that our system is definitely faster.

User Interface Improvement The second lesson we learned from this work is related to which improvements should be carried out to the user interface for making the platform more appealing from the user's perspective. Users interviewed for judging the tool provided feedback that can be summarized in the following two issues:

- Contextual information into the aspect visualization: in this prototype we did not take into account the possibility of having different kind of users: Basic and Advanced. While basic users can be satisfied from a simple graphical information supporting the detection of the most interesting aspects, advanced users wanted to see detailed information associated with them, i.e. the polarity value, a summary of supporters and opponents associated with each aspect, etc. This functionality will

be included in the next version of the platform.
- Animate the evolution of single aspect: the second issue raised by the users was related to the impossibility of observing how each aspect *evolves* during the analysis of the data stream. In particular, a desiderata is the possibility of focusing on a single aspect and to observe how such aspect is judged through time. This feature has been considered as a valuable support for associating peaks of supporters or opponents based on contextual events that cannot be tracked through the proposed system.

The two issues brought to light from users' feedbacks will be used as a starting point for improving the infrastructure of the presented platform. Thus it will be possible to employ such a platform in a larger scale context with the aim of increasing its technological readiness level.

2.8 Conclusions and Future Work

Results reported in the previous sections revealed the feasibility of the proposed architecture and of the implement techniques. Even if aspect recognition procedure presented in Section 2.5 may *lack* of precision and recall due to the adopted unsupervised techniques, the results reported in Section 5.4 shows that the effectiveness of the system is comparable with the supervised systems participated in the SemEval challenges. Thus, few changes in the proposed approach could result in significantly better performances. Examples of actions focus on the improvement of precision that could be achieved by adding a semantic clustering phase in the parsing pipeline shown in Figure 2.3. Then, by detecting the semantic distances between extracted aspect might help to discard uncorrelated aspects that may not refer to the reviewed product. Recall values could be increased as well by applying less strict rules than the ones presented in Section 2.5. These possibilities will be taken into account for possible future developments.

Another important part of this work focuses on aspect and opinion polarization. Tables 2.4 and 2.5 shown that the presented technique works well during the polarity inference phase. These results suggest that by using an

aggregation of (i) general purpose sentiment lexicons and (ii) specific ones, the polarity evaluation phase is positively affected.

Concerning the overall architecture, the presented solution provides a wide range of functionalities that can be applied to provide useful facilities for both customers and producers. For instance, the three level-tree structure shown in Figure 2.2 can be used to produce both a flexible ranking system and an effective representation of each expressed opinion that can highlight specific qualities and problematics (Figure 2.7) of each reviewed product.

In the future, efforts will be focused on several different perspectives. The first one concerns the developing of semantic clustering approach for extracting aspects This way, search would be based on inserted words' semantics rather than their syntax. As a result, ranking products by *screen* would also organize *display* and *screen_resolution* aspects rather than discarding them because of their different form.

Other possible progresses regard the application of different aspect extraction techniques on the implemented framework, the refinement of the user interface described in Section 2.6 and a more detailed comparison between multiple domain-specific lexicon. This last perspective could result in interesting developments concerning the production of a domain-distance metric and the integration of fuzzy membership for unclassified reviews or automatic domain labeling. Once domain-specific lexicons have been produced, they can be used alongside aspect extraction techniques to give a score value to portion of texts which are not provided with that additional information. Such an application could be easily benchmarked with existing reviews.

Bibliography

Akbik, Alan and Alexander Löser (2012). "KrakeN: N-ary Facts in Open Information Extraction." In: *Proceedings of the Joint Workshop on Auto-*

matic Knowledge Base Construction and Web-scale Knowledge Extraction. AKBC-WEKEX '12. Montreal, Canada: Association for Computational Linguistics, pp. 52–56. URL: http://dl.acm.org/citation.cfm?id=2391200.2391210 (cit. on p. 35).

Aprosio, Alessio Palmero et al. (2015). "Supervised Opinion Frames Detection with RAID." In: *Semantic Web Evaluation Challenges - Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*. Ed. by Fabien Gandon et al. Vol. 548. Communications in Computer and Information Science. Springer, pp. 251–263. DOI: 10.1007/978-3-319-25518-7_22. URL: http://dx.doi.org/10.1007/978-3-319-25518-7_22 (cit. on p. 32).

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." In: *LREC*, pp. 2200–2204 (cit. on p. 36).

Barbosa, Luciano and Junlan Feng (2010). "Robust Sentiment Detection on Twitter from Biased and Noisy Data." In: *COLING (Posters)*, pp. 36–44 (cit. on pp. 31, 33).

Bast, Hannah and Elmar Haussmann (2013). "Open Information Extraction via Contextual Sentence Decomposition." In: *Proceedings of the 2013 IEEE 7th International Conference on Semantic Computing*. ICSC '13. IEEE. Irvine, CA, USA (cit. on p. 35).

Cambria, E. and A. Hussain (2012). "Sentic album: Content-, concept-, and context-based online personal photo management system." In: *Cognitive Computation* 4.4, pp. 477–496 (cit. on p. 32).

Cambria, Erik (2016). "Affective Computing and Sentiment Analysis." In: *IEEE Intelligent Systems* 31.2, pp. 102–107. DOI: 10.1109/MIS.2016.31. URL: http://dx.doi.org/10.1109/MIS.2016.31 (cit. on pp. 28, 31).

Cambria, Erik and Amir Hussain (2015). *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Switzerland: Springer. ISBN: 978-3-319-23654-4 (cit. on p. 32).

Cambria, Erik, Soujanya Poria, et al. (2016). "SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives." In: *COLING*. ACL, pp. 2666–2677 (cit. on p. 36).

Chaturvedi, Iti, Erik Cambria, and David Vilares (2016). "Lyapunov filtering of objectivity for Spanish Sentiment Model." In: *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July*

24-29, 2016. IEEE, pp. 4474–4481. DOI: 10.1109/IJCNN.2016.7727785. URL: https://doi.org/10.1109/IJCNN.2016.7727785 (cit. on p. 32).

Chen, Danqi and Christopher D Manning (2014). "A Fast and Accurate Dependency Parser using Neural Networks." In: *Empirical Methods in Natural Language Processing (EMNLP)* (cit. on p. 40).

Chen, Tao et al. (2016). "Learning User and Product Distributed Representations Using a Sequence Model for Sentiment Analysis." In: *IEEE Comp. Int. Mag.* 11.3, pp. 34–44. DOI: 10.1109/MCI.2016.2572539. URL: https://doi.org/10.1109/MCI.2016.2572539 (cit. on p. 32).

Choi, Yejin and Claire Cardie (2010). "Hierarchical Sequential Learning for Extracting Opinions and Their Attributes." In: *Proceedings of the ACL 2010 Conference Short Papers*. ACLShort '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 269–274. URL: http://dl.acm.org/citation.cfm?id=1858842.1858892 (cit. on p. 34).

Clark, Kevin and Christopher D. Manning (2015). "Entity-Centric Coreference Resolution with Model Stacking." In: *Association for Computational Linguistics (ACL)* (cit. on p. 40).

Costa Pereira, Célia da, Mauro Dragoni, and Gabriella Pasi (2009). "A Prioritized "And" Aggregation Operator for Multidimensional Relevance Assessment." In: *AI*IA 2009: Emergent Perspectives in Artificial Intelligence, XIth International Conference of the Italian Association for Artificial Intelligence, Reggio Emilia, Italy, December 9-12, 2009, Proceedings*. Ed. by Roberto Serra and Rita Cucchiara. Vol. 5883. Lecture Notes in Computer Science. Springer, pp. 72–81. DOI: 10.1007/978-3-642-10291-2_8. URL: http://dx.doi.org/10.1007/978-3-642-10291-2_8 (cit. on p. 32).

Del Corro, Luciano and Rainer Gemulla (2013). "ClausIE: Clause-based Open Information Extraction." In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, pp. 355–366. ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488420. URL: http://doi.acm.org/10.1145/2488388.2488420 (cit. on p. 35).

Deng, Lingjia and Janyce Wiebe (2015). "MPQA 3.0: An Entity/Event-Level Sentiment Corpus." In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. Ed. by Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar. The Association for Computational Linguistics, pp. 1323–1328. ISBN: 978-1-941643-49-5. URL: http://aclweb.org/anthology/N/N15/N15-1146.pdf (cit. on p. 36).

Dragoni, M. (2015). "SHELLFBK: An Information Retrieval-based System For Multi-Domain Sentiment Analysis." In: *Proceedings of the 9th International Workshop on Semantic Evaluation*. SemEval '2015. Denver, Colorado: Association for Computational Linguistics, pp. 502–509 (cit. on p. 32).

Dragoni, Mauro, Antonia Azzini, and Andrea Tettamanzi (2010). "A Novel Similarity-Based Crossover for Artificial Neural Network Evolution." In: *Parallel Problem Solving from Nature - PPSN XI, 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*. Ed. by Robert Schaefer et al. Vol. 6238. Lecture Notes in Computer Science. Springer, pp. 344–353. DOI: 10.1007/978-3-642-15844-5_35. URL: http://dx.doi.org/10.1007/978-3-642-15844-5_35 (cit. on p. 34).

Dragoni, Mauro, Andrea G. B. Tettamanzi, and Célia da Costa Pereira (2014). "A Fuzzy System for Concept-Level Sentiment Analysis." In: *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*. Ed. by Valentina Presutti et al. Vol. 475. Communications in Computer and Information Science. Springer, pp. 21–27. DOI: 10.1007/978-3-319-12024-9_2. URL: http://dx.doi.org/10.1007/978-3-319-12024-9_2 (cit. on p. 32).

Dragoni, Mauro, Andrea G.B. Tettamanzi, and Célia da Costa Pereira (2015). "Propagating and Aggregating Fuzzy Polarities for Concept-Level Sentiment Analysis." English. In: *Cognitive Computation* 7.2, pp. 186–197. ISSN: 1866-9956. DOI: 10.1007/s12559-014-9308-6. URL: http://dx.doi.org/10.1007/s12559-014-9308-6 (cit. on p. 32).

Dragoni, Mauro, Andrea Tettamanzi, and Célia da Costa Pereira (2016). "DRANZIERA: An Evaluation Protocol For Multi-Domain Opinion Mining." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Ed. by Nicoletta Calzolari (Conference Chair) et al. Portorož, Slovenia: European Language Resources Association (ELRA). ISBN: 978-2-9517408-9-1 (cit. on p. 57).

Fader, Anthony, Stephen Soderland, and Oren Etzioni (2011). "Identifying Relations for Open Information Extraction." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 1535–1545. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145596 (cit. on p. 35).

Falke, Tobias et al. (2016). "Porting an Open Information Extraction System from English to German." In: *Proceedings of the 2016 Conference on*

Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pp. 892–898. URL: http://aclweb.org/anthology/D/D16/D16-1086.pdf (cit. on p. 35).

Fan, Teng-Kai and Chia-Hui Chang (2010). "Sentiment-oriented contextual advertising." In: *Knowl. Inf. Syst.* 23.3, pp. 321–344. DOI: 10.1007/s10115-009-0222-2. URL: http://dx.doi.org/10.1007/s10115-009-0222-2 (cit. on p. 34).

Fellbaum, Christiane (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press (cit. on pp. 39, 46).

Gamallo, Pablo and Marcos Garcia (2015). "Multilingual Open Information Extraction." In: *Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings*. Ed. by Francisco Pereira et al. Cham: Springer International Publishing, pp. 711–722. ISBN: 978-3-319-23485-4. DOI: 10.1007/978-3-319-23485-4_72. URL: http://dx.doi.org/10.1007/978-3-319-23485-4_72 (cit. on p. 35).

Gamallo, Pablo, Marcos Garcia, and Santiago Fern'andez-Lanza (2012). "Dependency-Based Open Information Extraction." In: *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*. Association for Computational Linguistics (cit. on p. 35).

Gangemi, Aldo, Valentina Presutti, and Diego Reforgiato Recupero (2014). "Frame-Based Detection of Opinion Holders and Topics: A Model and a Tool." In: *IEEE Comp. Int. Mag.* 9.1, pp. 20–30. DOI: 10.1109/MCI.2013.2291688. URL: https://doi.org/10.1109/MCI.2013.2291688 (cit. on p. 32).

Go, Alec, Richa Bhayani, and Lei Huang (2009). *Twitter Sentiment Classification using Distant Supervision*. CS224N Project Report, Standford University (cit. on pp. 31, 33).

Hu, Minqing and Bing Liu (2004). "Mining and summarizing customer reviews." In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*. Ed. by Won Kim et al. ACM, pp. 168–177. DOI: 10.1145/1014052.1014073. URL: http://doi.acm.org/10.1145/1014052.1014073 (cit. on pp. 28, 50–55, 57).

Jakob, Niklas and Iryna Gurevych (2010). "Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields." In: *Proceedings of the 2010 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp. 1035–1045. URL: http://www.aclweb.org/anthology/D10-1101 (cit. on pp. 31, 34).

Jin, Wei and Hung Hay Ho (2009). "A Novel Lexicalized HMM-based Learning Framework for Web Opinion mining." In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: ACM, pp. 465–472. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553435. URL: http://doi.acm.org/10.1145/1553374.1553435 (cit. on p. 34).

Jin, Wei, Hung Hay Ho, and Rohini K. Srihari (2009). "OpinionMiner: a novel machine learning system for web opinion mining and extraction." In: *KDD*, pp. 1195–1204 (cit. on pp. 31, 34).

Hinrichs, Erhard W. and Dan Roth, eds. (2003). *Accurate Unlexicalized Parsing*, pp. 423–430 (cit. on p. 40).

Kristina Toutanova Dan Klein, Christopher Manning and Yoram Singer (2003). "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network." In: *In Proceedings of HLT-NAACL 2003*, pp. 252–259 (cit. on p. 40).

Li, Fangtao, Minlie Huang, and Xiaoyan Zhu (2010). "Sentiment Analysis with Global Topics and Local Dependency." In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1913 (cit. on p. 34).

Liu, Bing, Minqing Hu, and Junsheng Cheng (2005). "Opinion observer: analyzing and comparing opinions on the Web." In: *WWW*, pp. 342–351 (cit. on pp. 31, 34).

Liu, Bing and Lei Zhang (2012). "A Survey of Opinion Mining and Sentiment Analysis." In: *Mining Text Data*. Ed. by C. C. Aggarwal and C. X. Zhai. Springer, pp. 415–463 (cit. on p. 31).

Liu, Qian et al. (2015). "Automated Rule Selection for Aspect Extraction in Opinion Mining." In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Qiang Yang and Michael Wooldridge. AAAI Press, pp. 1291–1297. URL: http://ijcai.org/Abstract/15/186 (cit. on p. 51).

Manning, Christopher D. et al. (2014). "The Stanford CoreNLP Natural Language Processing Toolkit." In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010 (cit. on pp. 39, 42).

Mausam et al. (2012). "Open Language Learning for Information Extraction." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 523–534. URL: http://dl.acm.org/citation.cfm?id=2390948.2391009 (cit. on p. 35).

Mei, Qiaozhu et al. (2007). "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs." In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: ACM, pp. 171–180. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242596. URL: http://doi.acm.org/10.1145/1242572.1242596 (cit. on p. 34).

Mitchell, Margaret et al. (2013). "Open Domain Targeted Sentiment." In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pp. 1643–1654. URL: http://aclweb.org/anthology/D/D13/D13-1171.pdf (cit. on p. 34).

Mukherjee, Arjun and Bing Liu (2012). "Aspect Extraction Through Semi-supervised Modeling." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. ACL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 339–348. URL: http://dl.acm.org/citation.cfm?id=2390524.2390572 (cit. on p. 34).

Oneto, Luca et al. (2016). "Statistical Learning Theory and ELM for Big Social Data Analysis." In: *IEEE Comp. Int. Mag.* 11.3, pp. 45–55. DOI: 10.1109/MCI.2016.2572540. URL: https://doi.org/10.1109/MCI.2016.2572540 (cit. on p. 33).

Pang, Bo and Lillian Lee (2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts." In: *ACL*, pp. 271–278 (cit. on p. 31).

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." In: *Proceedings of EMNLP*. Philadelphia: Association for Computational Linguistics, pp. 79–86 (cit. on pp. 28, 31).

Petrucci, Giulio and Mauro Dragoni (2015). "An Information Retrieval-Based System for Multi-domain Sentiment Analysis." In: *Semantic Web Evaluation Challenges - Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*. Ed. by Fabien Gandon et al. Vol. 548. Communications in Computer and Information Science. Springer, pp. 234–243. DOI: 10.1007/978-3-319-25518-7_20. URL: http://dx.doi.org/10.1007/978-3-319-25518-7_20 (cit. on p. 32).

P.J, Stone, D.C. Dunphy, and S. Marshall (1966). *The General Inquirer: A Computer Approach to Content Analysis*. Oxford, England: M.I.T. Press (cit. on p. 36).

Poria, Soujanya, Erik Cambria, and Alexander F. Gelbukh (2016). "Aspect extraction for opinion mining with a deep convolutional neural network." In: *Knowl.-Based Syst.* 108, pp. 42–49. DOI: 10.1016/j.knosys.2016.06.009. URL: https://doi.org/10.1016/j.knosys.2016.06.009 (cit. on pp. 32, 33).

Poria, Soujanya, Erik Cambria, Devamanyu Hazarika, et al. (2016). "A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks." In: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. Ed. by Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad. ACL, pp. 1601–1612. URL: http://aclweb.org/anthology/C/C16/C16-1151.pdf (cit. on p. 32).

Qiu, Guang et al. (2011). "Opinion Word Expansion and Target Extraction through Double Propagation." In: *Computational Linguistics* 37.1, pp. 9–27. DOI: 10.1162/coli_a_00034. URL: http://dx.doi.org/10.1162/coli_a_00034 (cit. on pp. 50–55, 57).

Quirk, Randolph et al. (1985). *A comprehensive grammar of the English language*. Vol. 397. Cambridge Univ Press (cit. on p. 38).

Recupero, Diego Reforgiato et al. (2015). "Sentilo: Frame-Based Sentiment Analysis." In: *Cognitive Computation* 7.2, pp. 211–225. DOI: 10.1007/s12559-014-9302-z. URL: https://doi.org/10.1007/s12559-014-9302-z (cit. on p. 32).

Riloff, Ellen, Siddharth Patwardhan, and Janyce Wiebe (2006). "Feature Subsumption for Opinion Analysis." In: *EMNLP*, pp. 440–448 (cit. on p. 32).

Sklar, Max and Kristian J. Concepcion (2014). "Timely Tip Selection for Foursquare Recommendations." In: *Poster Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014, Foster City, Silicon Valley, CA, USA, October 6-10, 2014*. Ed. by Li Chen and Jalal Mahmud. Vol. 1247. CEUR Workshop Proceedings. CEUR-WS.org. URL: http://ceur-ws.org/Vol-1247/recsys14_poster18.pdf (cit. on p. 34).

Socher, Richard et al. (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank." In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, WA: Association for Computational Linguistics, pp. 1631–1642 (cit. on p. 32).

Stanovsky, Gabriel et al. (2016). "Getting More Out Of Syntax with PropS." In: *CoRR* abs/1603.01648. URL: http://arxiv.org/abs/1603.01648 (cit. on p. 35).

Su, Qi et al. (2008). "Hidden sentiment association in chinese web opinion mining." In: *WWW*, pp. 959–968 (cit. on pp. 31, 34).

Thelwall, Mike et al. (2010). "Sentiment in short strength detection informal text." In: *JASIST* 61.12, pp. 2544–2558. DOI: 10.1002/asi.21416. URL: http://dx.doi.org/10.1002/asi.21416 (cit. on p. 33).

Titov, Ivan and Ryan McDonald (2008). "A Joint Model of Text and Aspect Ratings for Sentiment Summarization." In: *PROC. ACL-08: HLT*. Pp. 308–316 (cit. on p. 34).

Wang, Mingyin, Lei Li, and Fang Huang (2014). "Semi-supervised Chinese Open Entity Relation Extraction." In: *Proceedings of the 3rd IEEE International Conference on Cloud Computing and Intelligence Systems*. IEEE. Shenzhen and Hongkong, China (cit. on p. 35).

Wang, Q. F. et al. (2013). "Common sense knowledge for handwritten Chinese recognition." In: *Cognitive Computation* 5.2, pp. 234–242 (cit. on p. 32).

Wilson, Theresa, Janyce Wiebe, and Rebecca Hwa (2006). "Recognizing Strong and Weak Opinion Clauses." In: *Computational Intelligence* 22.2, pp. 73–99 (cit. on p. 32).

Wu, Fei and Daniel S. Weld (2010). "Open Information Extraction Using Wikipedia." In: *Proceedings of the 48th Annual Meeting of the Association*

for Computational Linguistics. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 118–127. URL: http://dl.acm.org/citation.cfm?id=1858681.1858694 (cit. on p. 35).

Wu, Yuanbin et al. (2009). "Phrase Dependency Parsing for Opinion Mining." In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pp. 1533–1541. URL: http://www.aclweb.org/anthology/D09-1159 (cit. on pp. 31, 34).

Yates, Alexander et al. (2007). "TextRunner: Open Information Extraction on the Web." In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. NAACL-Demonstrations '07. Rochester, New York: Association for Computational Linguistics, pp. 25–26. URL: http://dl.acm.org/citation.cfm?id=1614164.1614177 (cit. on p. 35).

Zhang, Meishan, Yue Zhang, and Duy-Tin Vo (2015). "Neural Networks for Open Domain Targeted Sentiment." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. Ed. by Lluís Márquez et al. The Association for Computational Linguistics, pp. 612–621. URL: http://aclweb.org/anthology/D/D15/D15-1073.pdf (cit. on p. 34).

Zhila, Alisa and Alexander F Gelbukh (2014). "Open Information Extraction for Spanish Language based on Syntactic Constraints." In: *ACL (Student Research Workshop)*, pp. 78–85 (cit. on p. 35).

3 Social Media Monitoring for Companies: A 4W Summarization Approach

Andi Rexha,
Mark Kröll,
Roman Kern

Abstract

Monitoring (social) media represents one means for companies to gain access to knowledge about, for instance, competitors, products as well as markets. As a consequence, social media monitoring tools have been gaining attention to handle amounts of data nowadays generated in social media. These tools also include summarisation services. However, most summarisation algorithms tend to focus on (i) first and last sentences respectively or (ii) sentences containing keywords.

In this work we approach the task of summarisation by extracting 4W (who, when, where, what) information from (social) media texts. Presenting 4W information allows for a more compact content representation than traditional summaries. In addition, we depart from mere named entity recognition (NER) techniques to answer these four question types by including non-rigid designators, i.e. expressions which do not refer to the same thing in all possible worlds such as "at the main square" or "leaders of political parties". To do that, we employ dependency parsing to identify grammatical characteristics for each question type. Every sentence is then represented as a 4W

block. We perform two different preliminary studies: selecting sentences that better summarise texts by achieving an F1-measure of 0.343, as well as a 4W block extraction for which we achieve F1-measures of 0.932; 0.900; 0.803; 0.861 for "who", "when", "where" and "what" category respectively.

In a next step the 4W blocks are ranked by relevance. The top three ranked blocks, for example, then constitute a summary of the entire textual passage. The relevance metric can be customised to the user's needs, for instance, ranked by up-to-dateness where the sentences' tense is taken into account. In a user study we evaluate different ranking strategies including (i) up-to-dateness, (ii) text sentence rank, (iii) selecting the firsts and lasts sentences or (iv) coverage of named entities, i.e. based on the number of named entities in the sentence.

Our 4W summarisation method presents a valuable addition to a company's (social) media monitoring toolkit, thus supporting decision making processes.

3.1 Introduction

Monitoring (social) media represents one means for companies collect insights about business relevant influence factors. This might be knowledge about their competitors and their products as well as prospective markets. As a consequence, social media monitoring tools have been gaining attention recently. These monitoring tools are able to handle amounts of data nowadays generated by social media services. The monitoring tools also provide summarisation services. Summarisation services act as a kind of filtering mechanism to identify and select only relevant parts instead of being overwhelmed with information. A lot of summarisation algorithms use simple strategies such as (i) focusing on first and, respectively, last sentences of a paragraph or (ii) selecting sentences containing keywords. These simple strategies are often successful in situations where a general gist of an article is required. However, these strategies do not allow for more customised summarisations, for instance, if a company is more interested in current affairs or the like.

In this work we approach the summarisation task by first extracting blocks of 4W (who, what, when, where) information from textual content and then rank them according to their relevance. Answering 4W question types relates to basic information-gathering and problem-solving activities. 4W blocks thus summarise the information in a natural language sentence allowing for a more compact content representation than traditional summaries where often the entire sentence is returned. In addition, we depart from mere named entity recognition (NER) techniques by including non-rigid designators, i.e. expressions which do not refer to the same thing in all possible worlds such as "at the main square" or "leaders of political parties". To give an example, from the following sentence, we first extract the underlined parts:

The armies of India and China would carry
a joint military exercise on counter-terrorism
on the borders of Pakistan in November this year.

We then assign these parts to the 4W question types resulting in the following 4W block:

who	the armies of India and China
what	a joint military exercise on counter-terrorism
when	in November this year
where	on the borders of Pakistan

To do that, we employ dependency parsing to identify grammatical characteristics for each question type. Similar to Open Information Extraction techniques, we generate tuples from the parse tree, especially for answering the "who" and the "what" question types. Every sentence is then represented as a 4W block. In the next step the 4W blocks are then ranked by relevance which equals the selection of the most relevant sentences. The top ranked blocks then constitute a summary of the entire textual passage. The relevance metric can be customised to the user's needs, for instance, ranked by up-to-dateness where the sentences' tense is taken into account. In a user study, we evaluate four different ranking strategies on a data set of news articles. The first ranking strategy considers only the first, the second and the last sentence as relevant similar, for instance, to the Lead

method (Baxendale, 1958). The second one is up-to-dateness taking into account temporal information. The third strategy is coverage, i.e. a ranking based on the number of Named Entities contained in the sentence, and the fourth one uses the TextSentenceRank algorithm (Seifert et al., 2013) which is an extension to the original TextRank algorithm (Mihalcea and Tarau, 2004).

We perform two different types of evaluation; sentence wise, achieving an F1-measure of 0.343 , and block wise achieving F1-measures of 0.932; 0.900; 0.803; 0.861 for "who", "when", "where" and "what" category respectively. Thus, our 4W summarisation method presents a valuable addition to a company's (social) media monitoring toolkit, for instance, to support decision making processes.

The next section of this discusses related work in the field of text summarisation. We describe implementation details in Section 3. In Section 4, we evaluate our approach on a data set of news articles. Lastly, we conclude the paper and present ideas for future work.

3.2 Related work

Text summarisation is the task of creating a representative short snippet of text out of a longer text, where the original text may be a single document or a collection of documents. In order to tackle this task two main families of methods have emerged: i) extractive text summarisation and ii) abstractive summarisation. For extractive summarisation fragments of the original text are selected and combined to form the final summary. Typically, the most representative sentences are selected – approaches vary in the way how these sentences are ranked. For example, the similarity of the sentences can be used to govern the selection of sentences (Erkan and Radev, 2004). The usage of graphs, based on the sentence structure, and applying the PageRank algorithm have been applied by Mihalcea and Tarau, 2004, and further refined by Seifert et al., 2013. Other extractive summarisation approaches apply optimisation algorithms based on the presence of concepts within the original text (Gillick and Favre, 2009). Other, more recent, approaches

study the usage of neural networks to learn to select the most representative sentences (Cao et al., 2015).

Abstractive text summarisation is less well researched, partially due to the good performance of the extractive summarisation methods. Here the methods try to detect the main underlying concepts and then to generate a summarisation by the use of text synthesis. Examples for abstractive summarisation are the based on sentence compression (Knight and Marcu, 2002) and using semantic graphs (F. Liu et al., 2018). Historically, the field has advanced driven by a number of initiatives: Most prominently the workshop series on Text Summarisation (WAS), the Document Understanding[1] (DUC) and the Text Analysis Conference[2] (TAC). In order to assess the quality of text summarisations, a number of evaluation metrics have been proposed, with Rouge, a Recall Oriented Understudy for Gisting Evaluation, (Lin and Och, 2004) being the most commonly used. Alternatives being actively researched as well, for example, taking more than one reference summarisation into account for more robust evaluation results (Hamid, Haraburda, and Tarau, 2016).

The term "Named Entities" is closely linked with the Message Understanding Conference (MUC) (Grishman and Sundheim, 1996). The task is defined to identify references to entities within text, where these entities usually represent persons, locations and organisations – the so called enamex classes. Later the task of Named Entity Recognition has also been extended to include references, dates or events. First approaches to this task are mainly based on hand crafted rules (Rau, 1991) and on lists of known named entities, usually referred to as gazetteer lists. In 2003 the CONLL workshop[3] extended the task to other languages and published a number of data-sets, which has been heavily used since then.

The most common approach to identify Named Entities are supervised methods, which rely on the availability of labelled training data in combination with learning algorithms. In this context, Hidden Markov Models have been successfully used (Bikel et al., 1997), as well as the more complex Conditional Random Fields (McCallum and Li, 2003). Apart from the main

[1] http://duc.nist.gov/
[2] http://www.nist.gov/tac/about/
[3] http://ifarm.nl/signll/conll/

algorithm, the approaches also differ in the way how features are generated. These features typically capture syntactic information, for example Part-Of-Speech (POS) tags. In addition, various knowledge bases have been studied for their usefulness to be used as features. In the AIDA system (Hoffart et al., 2011) one such knowledge base has been integrated, namely the YAGO (Suchanek, Kasneci, and Weikum, 2007), in order to improve the robustness of the extraction. Similarly, the task of named entity recognition has been extended to include text, which is not expected to be grammatically correct, e.g. from resources such as Twitter (X. Liu et al., 2011). Supervised methods require a fully annotated training corpus, which is tedious to create, therefore a number of approaches have been proposed to alleviate this problem, ranging from semi-supervised (or weakly-supervised) methods to unsupervised methods. For example, Pasca et al., 2006 propose a system to make use of the Web as resource and to start with a hand-crafted list of named entities as seeds. Following this approach does not require the usage of text processing tools, like syntactic parsers or gazetteer lists.

The aim of Open Information Extraction (OIE), introduced by Etzioni et al., 2008 is to find semantic relations from text expressed in a natural way. Different approaches have been proposed differing from the parsing techniques as well as from the used method (supervised or unsupervised) to extract those relations. Text Runner (Etzioni et al., 2008) uses a shallow parsing and trains a Bayes classifier on a small dataset. Other OIE tools that use shallow parsing are Reverb (Fader, Soderland, and Etzioni, 2011), which uses constraints based on the lexical information, and WOE-pos (Wu and Weld, 2010) which uses a supervised method on a high quality training data. Other tools like WOE-parse (Wu and Weld, 2010) and Ollie (Schmitz et al., 2012) use deep parsing. Both train classification models to learn patterns on the dependency tree. ClausIE (Del Corro and Gemulla, 2013) uses a totally different approach. It exploits the language knowledge for extracting propositions. The algorithms using deep parsing tend to outperform in terms of accuracy the ones using shallow parsing but do suffer on the speed of the extraction.

3.3 Approach

Extracting 4W blocks from (social) media and text in general requires some design choices. Different questions arise for creating and showing the information to the user:

- How do we extract the 4W blocks?
- How much information do we show to the user?
- Do we provide all the possible extracted 4W blocks?

In our view, 4W blocks represent a piece of information that contains a semantic relation between the "who", "what", "when" and "where". Sentences are a syntactic construct that semantically connect the phrases present in them. They are used as the source for extracting our 4W blocks. As a second design decision we select only blocks from a subset of sentences, by trying to select those that better summarise the content of the whole text.

The designed system can be described in the following two steps:

- Extract the components of the 4W blocks for each sentence.
- Rank the 4W blocks (= sentences) according to their relevance.

In the following subsections, we explain each of these steps.

3.3.1 Extracting the 4W blocks

In this work we focus on (social) media containing text expressed in a grammatically correct form. This is an important clue for pre-processing and annotating it with syntactic information. We use the Stanford CoreNLP (Manning et al., 2014) to annotate the sentences, words, Part-Of-Speech (POS) tagging, Named Entities (NE), Temporal Expressions (TE) and the grammatical dependency of each sentence of the text. To acquire information about the semantic of each sentence, we focus on extracting propositions in form of a triple:

Example {subject, predicate, object}

From the propositions we can map naturally the "who" to the "subject" and the "what" to the "predicate". The "when" and "where" are then part of the "object" part of the proposition.

These propositions can be extracted by using the Open Information Extraction (OIE) tool called ClausIE (Del Corro and Gemulla, 2013). Given a sentence, the tool produces all the possible predicates that are contained in it. Consider Example 1:

"Officials stated that the killer was not in town on Saturday!"

The produced propositions from the algorithm are:

- {"Officials", "stated", "that the killer was not in town on Saturday"}
- {"the killer", "was not", "in town on Saturday"}

As we can see, there are different options for selecting the sentences. Some of the propositions are contained in others. There can be also disjoint propositions. Consider Example 2:

"We went to Italy and we also played football."

The resulting propositions are:

- {"We", "went", "to Italy"}
- {"we", "played", "football also"}
- {"we", "played", "football"}

We decided to present only a subset of the blocks extracted for each sentence. For selecting the best candidates we propose following three strategies:

1. Longest proposition: we compare the proposition by the number of words. In Example 1 this strategy would have selected the first option.
2. Shortest proposition that doesn't lose Named Entities and Temporal Information: We select the shortest proposition that contains named entities or temporal information. In the Example 2 it would have selected the first option.

3. Shortest proposition selects the proposition which contains the fewest number of words.

After having selected a 4W block for the sentence, we still need to map the "when" and "where" with the "object" of the proposition. From the "object", we select the Named Entities as a "where", and Temporal Expression as "when". We extract these expressions by applying the CoreNLP annotation tool. As a further step, in order to find phrases that aren't Named Entities (for example "at the restaurant" in Example 2) or Temporal Expressions we use prepositions including "in", "at", "on", "next to", "under", "below", "over", "above", etc.

3.3.2 Ranking the 4W blocks

Ranking the previously extracted 4W blocks equals selecting the most relevant sentences from a given text. In this work we experiment with and evaluate four different ranking strategies; some of them are inspired by literature such as the Lead method from Baxendale, 1958 who noticed that relevant information is often placed at the beginning and at the ending of a text.

1. First/last: 4W blocks are selected from the first sentences and the last one. (inspired by the Lead method (Baxendale, 1958)).
2. Coverage: Select 4W blocks with the most question types answered (where all fields of the block are filled).
3. Up-to-dateness: Select 4W blocks where the corresponding sentence's tense is either the present or future tense.
4. TextSentenceRank: an unsupervised approach for identifying key sentence; an extension to the original TextRank algorithm.

3.4 Evaluation

For evaluating the current system, we have two different methods: sentence wise and block-wise. We selected a dataset composed of 53 annotated news

articles, manually annotated. In Table 1 we show the precision, recall and F1 measure for each of the strategies describe above:

	Precision	Recall	F1 measure
Coverage based	0.015	0.028	0.019
Up-to-dateness	0.030	0.056	0.039
Lead strategy	0.263	0.490	0.343
Text sentence rank	0.157	0.303	0.207

Table 3.1: Evaluation the different strategies for text summarisation.

As we can see from the Table 3.2, the Lead strategy outperforms the other strategies. For the block wise evaluation, we need to define different metrics. Matching annotation can be also partial, which we want to consider them too. Consider following extractions:

"who": "The prime minister" vs. "who": "prime minister".

Although these two extractions do not match in length, they exhibit the same semantic information. In order to take all possible matching variations into account, we define following matching types:

- correct match: The start and end offset of the annotation in the evaluation document match the offsets of the ground truth fully.
- partial match: The span of the annotation in the evaluation document only overlaps the annotation in the ground truth document.
- missing match: An annotation in the ground truth does not have a correct, partial or incorrect match in the evaluation document.
- spurious match: An annotation in the evaluation document does not have a correct, partial or incorrect match in the ground truth document.

We define the metrics for our evaluation as follow:

- precision = (correct + 0.5 * partial) / (correct + spurious + partial)
- recall = (correct + 0.5 * partial) / (correct + missing + partial)
- f-measure = (3* precision * recall) / (2*recall + precision)

For the evaluation of the information blocks we have selected a set of 40 sentences. We present the results that we achieved by using a combination of the strategies selecting the 4W blocks in Table 2.

	Precision	Recall	F1 measure
Who	0.950	0.900	0.932
What	0.861	0.861	0.861
When	0.900	0.900	0.900
Where	0.805	0.800	0.803

Table 3.2: Evaluation of the each "W" in the extracted 4W blocks from a set of 40 sentences.

For all question types (4Ws) we achieve an F1-measure higher than 0.8 which is a very promising result. We believe these high values are due to the natural way of mapping 4W blocks to propositions.

3.5 Conclusion

In this paper we have proposed a novel approach for monitoring (social) media by summarising texts using a 4W block ("who", "what", "where" and "when"). We have combined text summarisation approaches with Open Information Extraction techniques. We performed an evaluation on a small dataset of 53 news articles showing preliminary results, on one hand, promising ones for the extraction of 4W blocks, and on the other hand, rather low ones for the identification of relevant sentences.

In the future works we intend to analyse different dataset by considering multiple users' annotations and their inter-rating agreements. We also intend to define a notion of minimal information, which might help compressing information contained in sentences. Finally, we plan to apply the same techniques also for other languages. Our primary target will be the German language.

Bibliography

Baxendale, Phyllis B (1958). "Machine-made index for technical literature—an experiment." In: *IBM Journal of research and development* 2.4, pp. 354–361 (cit. on pp. 72, 77).

Bikel, Daniel M et al. (1997). "Nymble: a high-performance learning name-finder." In: *Proceedings of the fifth conference on Applied natural language processing*. Association for Computational Linguistics, pp. 194–201 (cit. on p. 73).

Cao, Ziqiang et al. (2015). "Ranking with recursive neural networks and its application to multi-document summarization." In: *Twenty-ninth AAAI conference on artificial intelligence* (cit. on p. 73).

Del Corro, Luciano and Rainer Gemulla (2013). "Clausie: clause-based open information extraction." In: *Proceedings of the 22nd international conference on World Wide Web*, pp. 355–366 (cit. on pp. 74, 76).

Erkan, Günes and Dragomir R Radev (2004). "Lexrank: Graph-based lexical centrality as salience in text summarization." In: *Journal of artificial intelligence research* 22, pp. 457–479 (cit. on p. 72).

Etzioni, Oren et al. (2008). "Open information extraction from the web." In: *Communications of the ACM* 51.12, pp. 68–74 (cit. on p. 74).

Fader, Anthony, Stephen Soderland, and Oren Etzioni (2011). "Identifying relations for open information extraction." In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pp. 1535–1545 (cit. on p. 74).

Gillick, Dan and Benoit Favre (2009). "A scalable global model for summarization." In: *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pp. 10–18 (cit. on p. 72).

Grishman, Ralph and Beth M Sundheim (1996). "Message understanding conference-6: A brief history." In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics* (cit. on p. 73).

Hamid, Fahmida, David Haraburda, and Paul Tarau (2016). "Evaluating Text Summarization Systems with a Fair Baseline from Multiple Reference Summaries." In: *European Conference on Information Retrieval*. Springer, pp. 351–365 (cit. on p. 73).

Hoffart, Johannes et al. (2011). "Robust disambiguation of named entities in text." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 782–792 (cit. on p. 74).

Knight, Kevin and Daniel Marcu (2002). "Summarization beyond sentence extraction: A probabilistic approach to sentence compression." In: *Artificial Intelligence* 139.1, pp. 91–107 (cit. on p. 73).

Lin, Chin-Yew and FJ Och (2004). "Looking for a few good metrics: ROUGE and its evaluation." In: *Ntcir Workshop* (cit. on p. 73).

Liu, Fei et al. (2018). "Toward abstractive summarization using semantic representations." In: *arXiv preprint arXiv:1805.10399* (cit. on p. 73).

Liu, Xiaohua et al. (2011). "Recognizing named entities in tweets." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 359–367 (cit. on p. 74).

Manning, Christopher D et al. (2014). "The Stanford CoreNLP natural language processing toolkit." In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60 (cit. on p. 75).

McCallum, Andrew and Wei Li (2003). "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons." In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 188–191 (cit. on p. 73).

Mihalcea, Rada and Paul Tarau (2004). "Textrank: Bringing order into text." In: *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411 (cit. on p. 72).

Pasca, Marius et al. (2006). "Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge." In: *AAAI*. Vol. 6, pp. 1400–1405 (cit. on p. 74).

Rau, Lisa F (1991). "Extracting company names from text." In: *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*. Vol. 1. IEEE, pp. 29–32 (cit. on p. 73).

Schmitz, Michael et al. (2012). "Open language learning for information extraction." In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. Association for Computational Linguistics, pp. 523–534 (cit. on p. 74).

Seifert, Christin et al. (2013). "Text Representation for Efficient Document Annotation." In: *J. UCS* 19.3, pp. 383–405 (cit. on p. 72).

Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum (2007). "Yago: a core of semantic knowledge." In: *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706 (cit. on p. 74).

Wu, Fei and Daniel S Weld (2010). "Open information extraction using Wikipedia." In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, pp. 118–127 (cit. on p. 74).

4 An Embedding Approach For Microblog Polarity Classification

Andi Rexha,
Mark Kröll,
Roman Kern,
Mauro Dragoni

Abstract

In the last years, forms of communication such as social media have emerged. Short and unstructured messages are used to share, interact and collaborate in different online communities. Identifying the nature of the emotion (positive or negative) expressed in these kind of text is a big challenge for the standard Natural Language Processing (NLP). Tweets are one of the most popular type of short messages. In this work we try to predict the polarity (positive or negative emotion) expressed by the user for a specific target phrase in a tweet. We try to exploit a Twitter Word2Vec[1] model in order to classify the polarity of these message refereed to the target phrase. We use two approaches to extract the features: windows-based and message-based. Evaluating with the *SemEval 2016 Task 4* dataset, we show that these simple approaches perform quite well, even though they do not use any polarity of single words. We also show that the performance of considering the whole tweet message is slightly better than the one considering a window around the target phrase.

[1]Word2Vec models provide a representation of words in a feature space that reflects their relation to other words in the training corpus

4.1 Introduction

With the growing popularity of the online social media, different forms of communication are being used more and more often. The trend of messaging is shifting to microblogging and short texts which usually are unstructured and very informal. While users of these social media aren't limited to specific type of text, they usually express their opinions or emotions about specific interests.

One of the most popular social media providing sharing of short texts is Twitter and its messages are called Tweets. The language used in these messages is very informal, with creative spelling and punctuation, misspellings, slang, URLs, and abbreviations. The difficulty in processing this kind of text is challenging for researchers.

Efforts have been made in tasks for automatically predicting sentiment polarity (whether positive or negative) of tweets. Even more challenging is to predict the opinion of specific target. In order to illustrate this, consider the following example:

*New features **@Microsoft** suck. Check them back! **#Linux** solutions are awesome*

There might be a neutral overall opinion, being that the first part "New features **@Microsoft** suck" expresses a negative emotion meanwhile the last part of the message "#Linux solutions are awesome." expresses a positive one. The two different references of these opinions (in this case *@Microsoft* and *#Linux*) are called the target phrases. In this paper we try to address exactly this challenge. Specifically we try to automatically predict the polarity (whether positive or negative) in a message about a given target. For instance, in the previous example, the algorithm should return a positive aspect about the target *@Microsoft* and a negative one about the target *#Linux*.

In order to tackle the challenge, in this paper we explore the semantic information given by a Word2Vec (Mikolov et al., 2013) trained model on twitter messages. Word2Vec models provide a representation of words in a feature space that reflects their relation to other words in the training corpus. We investigate whether the role of the target by using this technique has different outcomes compared to the only use of the words close to the target phrase. To evaluate this approach we use the test and golden standard dataset of the *Semeval 2016 Task #4* challenge about Twitter sentiment mining.

The paper is structure as follow. In the section 4.2 we present the approach of the challengers of the SemEval task of the 2015 and other related works. In the section 4.3 we suggest the different way to extract the features and in section 4.4 we compare the different approaches and learning algorithms. In the 4.5 section we draw the conclusion and the future work.

4.2 Related Work

The task of Sentiment Analysis, also known as opinion mining (Pang and Lee, 2008; Liu and Zhang, 2012), is to classify textual content according to expressed emotions and opinions. Sentiment classification has been a challenging topic in Natural Language Processing (Wiebe, Wilson, and Cardie, 2005). It is commonly defined as a binary classification task to assign a sentence either positive or negative polarity (Pang, Lee, and Vaithyanathan, 2002). Turney's work was among the first ones to tackle automatic sentiment classification (Turney, 2002). He employed an information-theoretic measure, i.e. mutual information, between a text phrase and the words "excellent" and "poor" as a decision metric.

The approaches presented above are applied at the document-level (M. Dragoni, 2015; Petrucci and Mauro Dragoni, 2015; Rexha et al., 2016a; Federici and Mauro Dragoni, 2016a), i.e., the polarity value is assigned to the entire document content. However, in some case, for improving the accuracy of the sentiment classification, a more fine-grained analysis of a document is needed. Hence, the sentiment classification of the single sentences, has to be performed. In the literature, we may find approaches ranging from the use of fuzzy logic (Mauro Dragoni, A. G. Tettamanzi,

and Costa Pereira, 2015; Mauro Dragoni, A. G. B. Tettamanzi, and Costa Pereira, 2014; Petrucci and Mauro Dragoni, 2016) to the use of aggregation techniques (Costa Pereira, Mauro Dragoni, and Pasi, 2009) for computing the score aggregation of opinion words. In the case of sentence-level sentiment classification, two different sub-tasks have to be addressed: (i) to determine if the sentence is subjective or objective, and (ii) in the case that the sentence is subjective, to determine if the opinion expressed in the sentence is positive, negative, or neutral. The task of classifying a sentence as subjective or objective, called "subjectivity classification", has been widely discussed in the literature (Federici and Mauro Dragoni, 2016b; Riloff, Patwardhan, and Wiebe, 2006; Wilson, Wiebe, and Hwa, 2006) and systems implementing the capabilities of identifying opinion's holder, target, and polarity have been presented (Aprosio et al., 2015). Once subjective sentences are identified, the same methods as for sentiment classification may be applied. For example, in Hatzivassiloglou and Wiebe, 2000 the authors consider gradable adjectives for sentiment spotting; while in Kim and Hovy, 2007; and Rexha et al., 2016b the authors built models to identify some specific types of opinions.

A particular attention should be given also to the application of sentiment analysis in social networks (Mauro Dragoni, 2017; Mauro Dragoni, Costa Pereira, et al., 2016). Micro-blogging data such as tweets differs from regular text as it is extremely noisy, informal and does not allow for long messages (which might not be a disadvantage (Bermingham and Smeaton, 2010). As a consequence, analyzing sentiment in Twitter data poses a lot of opportunities. Traditional feature representations such as part-of-speech information or the usage of lexicon features such as SentiWordNet have to be re-evaluated in the light of Twitter data. In case of part-of-speech information, Gimpel et al., 2011 annotated tweets and developed a tagset and features to train an adequate tagger. Kouloumpis et al. (Kouloumpis, Wilson, and Moore, 2011) investigated the usefulness of existing lexical resources and other features including part-of-speech information in the analysis task.

Go, Bhayani, and Huang, 2009, for instance, used emoticons as additional features, for example, ":)" and ":-)" for the positive class, ":(" and ":-(" for the negative class. They then applied machine learning techniques such as support vector machines to classify the tweets into a positive and a negative class. Agarwal, Biadsy, and Mckeown, 2009 introduced POS-specific prior

polarity features along with using a tree kernel for tweet classification. Barbosa and Feng, 2010 present a robust approach to Twitter sentiment analysis. The robustness is based on an abstract representation of tweets as well as the usage of noisy/biased labels from three websites to train their model.

Last but not least, recent years have seen a lot of participation in the annual SemEval tasks on Twitter Sentiment Analysis (Wilson, Kozareva, et al., 2013; Rosenthal, Ritter, et al., 2014; Rosenthal, Nakov, et al., 2015). This event provides optimal conditions to implement novel ideas and is a good starting point to catch up on the latest trends in this area.

4.3 Approach

As exlained in the previous sections we intend to experiment two different approaches for extracting mining features. In the first approach we use the sole information of each word without considering the position of the target phrase. On the other hand, in the second approach we consider only the surrounding of the target phrase. As a preprocessing step, we annotate the tweets (words, Part Of Speach Tagging etc.) by using the Tweet NLP library [2]. Further, for each word of the tweet we extract the Word2Vec vector representation by using a Twitter model trained over 400 million tweets [3]. In the postprocessing step we make an average over the considered segment (every word or words within the window). As the last step we use a binary feature which is set to 1 if in the tweet exists any negation word (don't, not, ...). We believe that this feature can give a hint to the learning algorithm whether the expressed emotion might be negated without considering those kind of words. Below we explain more in detail each of the approaches for extracting the features.

Whole Tweet Run The whole tweet run can be expained in the current steps:

[2] http://www.cs.cmu.edu/ãrk/TweetNLP/
[3] http://www.fredericgodin.com/software/

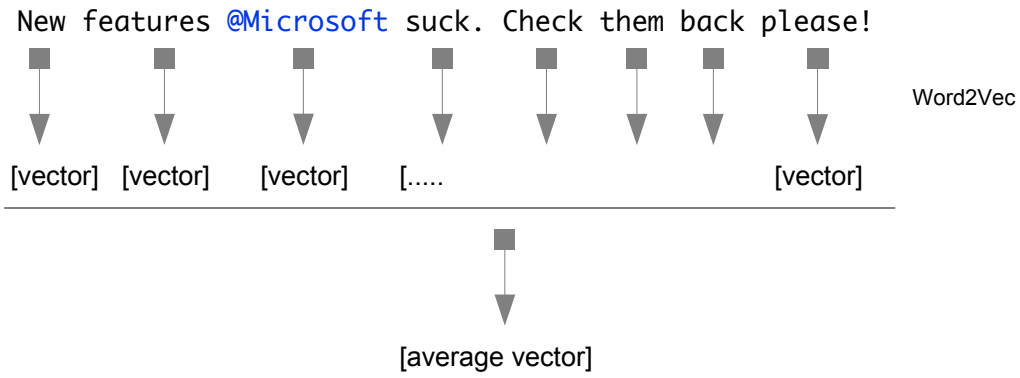


Figure 4.1: Whole Tweet Approach: showing how we extract the features without even considering the target phrase

- Preprocess the tweet messages, extracting each word
- For each of the words of the tweet, extract the Word2Vec vector
- For each corresponding feature extracted from the Word2Vec, make an average
- a binary feature as negation: if one of the words in the tweet contains a "not" or ends with a "'t", the feature is set to 1, otherwise, to 0

Figure 4.1 illustrates the way the Word2Vec vectors are combined to create the final vector representation for the *Whole Tweet approach*

Window Run The window run can be expained in the current steps:

- Preprocess the tweet messages, extracting each word
- Annotate the target of the tweet
- Build a window of "n" words from left and right of the target
- Extract the Word2Vec value for each word of the window
- For each corresponding feature extracted from the Word2Vec, make an average
- a binary feature as negation: if one of the words in the tweet contains a "not" or ends with a "'t", the feature is set to 1, otherwise, to 0

We show the extraction of the Word2Vec features for the *Window approach* in the figure 4.2.

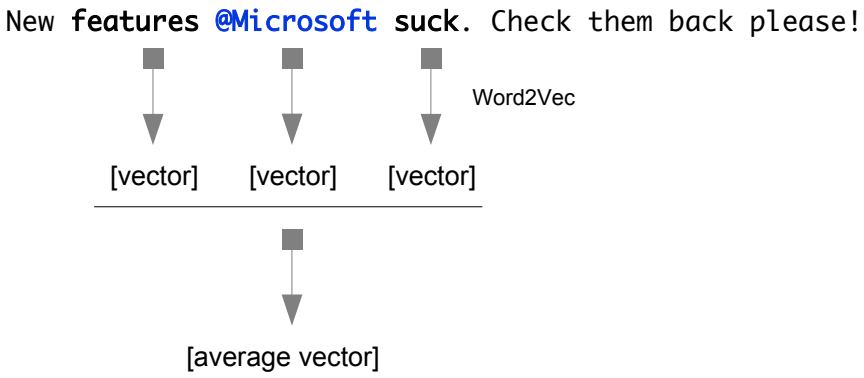


Figure 4.2: Window approach: Extracting the features from a window of words close to the target phrase (in the example the size of the window is 1)

4.4 Results

To evaluate the two different approaches that we proposed in the section 4.3 we have trained different classifiers to predict the opinion of the target phrase in a Tweet. We have used the dataset of the *Semeval 2016 Task 4* about sentiment analysis in Tweets. The training set is composed by 3858 entries and the evaluation set by 10551 entries. Both datasets are skewed. The training set contains 17% of negative and 83% of positive and the evaluation set of 22% of negative and 78% of positive examples. For each of the approaches we present the evaluation for the positive and negative classes by displaying the precision, recall and F1 measure. In the tables 4.1 and 4.2 we show the performance of the positive and negative classes for the full text approach. On the other hand, in the tables 4.3 and 4.4 we present the evaluation for the window approach. The chosen size of the window is set to 3. We believe that this size reflect the idea of chosing related words close to the target phrase.

Something to highlight from the tables is that the accuracy of the negative class is lower. We believe that this is due to the skewed nature of the dataset. Another detail to note is the difference between the two approaches. This characteristic might be due to the fact that we throw away some important

	Precision	Recall	F1-Measure
Naive Bayes	0.396	**0.733**	0.514
Support Vector Machine	**0.724**	0.347	0.469
Logistic Regression	0.606	0.481	**0.536**
Random Tree	0.321	0.266	0.291

Table 4.1: Evaluation for the Negative Opinions by using an average of the Word2Vec over all the words composing the tweet

	Precision	Recall	F1-Measure
Naive Bayes	**0.900**	0.681	0.775
Support Vector Machine	0.838	**0.962**	**0.896**
Logistic Regression	0.860	0.911	0.885
Random Tree	0.801	0.840	0.820

Table 4.2: Evaluation for the Positive Opinions by using an average of the Word2Vec over all the words composing the tweet

information that are not in the proximity to the target phrase.

4.5 Conclusion

In this paper we have evaluate an approach by using the semantic information collected from the Word2Vec for the prediction of the polarity in tweets. Specifically we have addressed the opinion mining for target phrases.

In future work we intend to exploit the effect of the dependency trees in tweets. The text proximity can give just partial information about the semantic proximity of the positive or negative words in the short messages. We believe that exploiting this information can improve the performance of the classifying algoritm.

	Precision	Recall	F1-Measure
Naive Bayes	0.303	**0.732**	**0.429**
Support Vector Machine	**0.442**	0.273	0.338
Logistic Regression	0.396	0.391	0.394
Random Tree	0.260	0.281	0.271

Table 4.3: Evaluation for the Negative Opinions by using a window of size 3

	Precision	Recall	F1-Measure
Naive Bayes	**0.872**	0.520	0.652
Support Vector Machine	0.813	**0.902**	**0.855**
Logistic Regression	0.827	0.830	0.829
Random Tree	0.791	0.773	0.781

Table 4.4: Evaluation for the Positive Opinions by using a window of size 3

Acknowledgment

This work is funded by the KIRAS program of the Austrian Research Promotion Agency (FFG) (project number 840824). The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labour and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG

Bibliography

Agarwal, Apoorv, Fadi Biadsy, and Kathleen R. Mckeown (2009). "Contextual Phrase-level Polarity Analysis Using Lexical Affect Scoring and Syntactic N-grams." In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '09 (cit. on p. 86).

Aprosio, Alessio Palmero et al. (2015). "Supervised Opinion Frames Detection with RAID." In: *Semantic Web Evaluation Challenges - Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*. Ed. by Fabien Gandon et al. Vol. 548. Communications in Computer and Information Science. Springer, pp. 251–263. DOI: 10.1007/978-3-319-25518-7_22. URL: http://dx.doi.org/10.1007/978-3-319-25518-7_22 (cit. on p. 86).

Barbosa, Luciano and Junlan Feng (2010). "Robust Sentiment Detection on Twitter from Biased and Noisy Data." In: *COLING (Posters)*, pp. 36–44 (cit. on p. 87).

Bermingham, Adam and Alan F. Smeaton (2010). "Classifying Sentiment in Microblogs: Is Brevity an Advantage?" In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. CIKM '10. ACM (cit. on p. 86).

Costa Pereira, Célia da, Mauro Dragoni, and Gabriella Pasi (2009). "A Prioritized "And" Aggregation Operator for Multidimensional Relevance Assessment." In: *AI*IA 2009: Emergent Perspectives in Artificial Intelligence, XIth International Conference of the Italian Association for Artificial Intelligence, Reggio Emilia, Italy, December 9-12, 2009, Proceedings*. Ed. by Roberto Serra and Rita Cucchiara. Vol. 5883. Lecture Notes in Computer Science. Springer, pp. 72–81. DOI: 10.1007/978-3-642-10291-2_8. URL: http://dx.doi.org/10.1007/978-3-642-10291-2_8 (cit. on p. 86).

Dragoni, M. (2015). "SHELLFBK: An Information Retrieval-based System For Multi-Domain Sentiment Analysis." In: *Proceedings of the 9th International Workshop on Semantic Evaluation*. SemEval '2015. Denver, Colorado: Association for Computational Linguistics, pp. 502–509 (cit. on p. 85).

Dragoni, Mauro (2017). "A Three-Phase Approach for Exploiting Opinion Mining in Computational Advertising." In: *IEEE Intelligent Systems* 32.3, pp. 21–27. DOI: 10.1109/MIS.2017.46. URL: https://doi.org/10.1109/MIS.2017.46 (cit. on p. 86).

Dragoni, Mauro, Célia da Costa Pereira, et al. (2016). "SMACk: An Argumentation Framework for Opinion Mining." In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. Ed. by Subbarao Kambhampati. IJCAI/AAAI Press, pp. 4242–4243. URL: http://www.ijcai.org/Abstract/16/641 (cit. on p. 86).

Dragoni, Mauro, Andrea G. B. Tettamanzi, and Célia da Costa Pereira (2014). "A Fuzzy System for Concept-Level Sentiment Analysis." In: *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*. Ed. by Valentina Presutti et al. Vol. 475. Communications in Computer and Information Science. Springer, pp. 21–27. DOI: 10.1007/978-3-319-12024-9_2. URL: http://dx.doi.org/10.1007/978-3-319-12024-9_2 (cit. on p. 86).

Dragoni, Mauro, Andrea G.B. Tettamanzi, and Célia da Costa Pereira (2015). "Propagating and Aggregating Fuzzy Polarities for Concept-Level Sentiment Analysis." English. In: *Cognitive Computation* 7.2, pp. 186–197. ISSN: 1866-9956. DOI: 10.1007/s12559-014-9308-6. URL: http://dx.doi.org/10.1007/s12559-014-9308-6 (cit. on p. 85).

Federici, Marco and Mauro Dragoni (2016a). "A Knowledge-Based Approach for Aspect-Based Opinion Mining." In: *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*. Ed. by Harald Sack et al. Vol. 641. Communications in Computer and Information Science. Springer, pp. 141–152. DOI: 10.1007/978-3-319-46565-4_11. URL: https://doi.org/10.1007/978-3-319-46565-4_11 (cit. on p. 85).

Federici, Marco and Mauro Dragoni (2016b). "Towards Unsupervised Approaches For Aspects Extraction." In: *Joint Proceedings of the 2th Workshop on Emotions, Modality, Sentiment Analysis and the Semantic Web and the 1st International Workshop on Extraction and Processing of Rich Semantics from Medical Texts co-located with ESWC 2016, Heraklion, Greece, May 29, 2016.* Ed. by Mauro Dragoni et al. Vol. 1613. CEUR Workshop Proceedings. CEUR-WS.org. URL: http://ceur-ws.org/Vol-1613/paper_2.pdf (cit. on p. 86).

Gimpel, Kevin et al. (2011). "Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*. HLT '11 (cit. on p. 86).

Go, Alec, Richa Bhayani, and Lei Huang (2009). *Twitter Sentiment Classification using Distant Supervision*. CS224N Project Report, Standford University (cit. on p. 86).

Hatzivassiloglou, Vasileios and Janyce Wiebe (2000). "Effects of Adjective Orientation and Gradability on Sentence Subjectivity." In: *COLING*, pp. 299–305 (cit. on p. 86).

Kim, Soo-Min and Eduard H. Hovy (2007). "Crystal: Analyzing Predictive Opinions on the Web." In: *EMNLP-CoNLL*, pp. 1056–1064 (cit. on p. 86).

Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore (2011). "Twitter Sentiment Analysis: The Good the Bad and the OMG!" In: *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Spain* (cit. on p. 86).

Liu, Bing and Lei Zhang (2012). "Mining Text Data." In: ed. by C. Charu Aggarwal and ChengXiang Zhai. Springer US. Chap. A Survey of Opinion Mining and Sentiment Analysis, pp. 415–463 (cit. on p. 85).

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781* (cit. on p. 85).

Pang, Bo and Lillian Lee (2008). "Opinion Mining and Sentiment Analysis." In: *Foundations and Trends in Information Retrieval* 2.1-2 (cit. on p. 85).

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 79–86. DOI: 10.3115/1118693.1118704. URL: http://dx.doi.org/10.3115/1118693.1118704 (cit. on p. 85).

Petrucci, Giulio and Mauro Dragoni (2015). "An Information Retrieval-Based System for Multi-domain Sentiment Analysis." In: *Semantic Web Evaluation Challenges - Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*. Ed. by Fabien Gandon et al. Vol. 548. Communications in Computer and Information Science. Springer, pp. 234–243. DOI: 10.1007/978-3-319-25518-7_20. URL: http://dx.doi.org/10.1007/978-3-319-25518-7_20 (cit. on p. 85).

Petrucci, Giulio and Mauro Dragoni (2016). "The IRMUDOSA System at ESWC-2016 Challenge on Semantic Sentiment Analysis." In: *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*. Ed. by Harald Sack et al. Vol. 641. Communications in Computer and Information Science. Springer, pp. 126–140. DOI: 10.1007/978-3-319-46565-4_10. URL: https://doi.org/10.1007/978-3-319-46565-4_10 (cit. on p. 86).

Rexha, Andi et al. (2016a). "Exploiting Propositions for Opinion Mining." In: *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*.

Ed. by Harald Sack et al. Vol. 641. Communications in Computer and Information Science. Springer, pp. 121–125. DOI: 10.1007/978-3-319-46565-4_9. URL: https://doi.org/10.1007/978-3-319-46565-4_9 (cit. on p. 85).

Rexha, Andi et al. (2016b). "Polarity Classification for Target Phrases in Tweets: A Word2Vec Approach." In: *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*. Ed. by Harald Sack et al. Vol. 9989. Lecture Notes in Computer Science, pp. 217–223. DOI: 10.1007/978-3-319-47602-5_40. URL: https://doi.org/10.1007/978-3-319-47602-5_40 (cit. on p. 86).

Riloff, Ellen, Siddharth Patwardhan, and Janyce Wiebe (2006). "Feature Subsumption for Opinion Analysis." In: *EMNLP*, pp. 440–448 (cit. on p. 86).

Rosenthal, Sara, Preslav Nakov, et al. (2015). "SemEval-2015 Task 10: Sentiment Analysis in Twitter." In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics (cit. on p. 87).

Rosenthal, Sara, Alan Ritter, et al. (2014). "SemEval-2014 Task 9: Sentiment Analysis in Twitter." In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University (cit. on p. 87).

Turney, Peter D. (2002). "Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews." In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics (cit. on p. 85).

Wiebe, Janyce, Theresa Wilson, and Claire Cardie (2005). "Annotating Expressions of Opinions and Emotions in Language." In: *Language Resources and Evaluation* 1.2 (cit. on p. 85).

Wilson, Theresa, Zornitsa Kozareva, et al. (2013). "Semeval-2013 task 2: Sentiment analysis in twitter." In: *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computation Linguistics (cit. on p. 87).

Wilson, Theresa, Janyce Wiebe, and Rebecca Hwa (2006). "Recognizing Strong and Weak Opinion Clauses." In: *Computational Intelligence* 22.2, pp. 73–99 (cit. on p. 86).

5 A Neural-based Architecture For Small Datasets Classification

Andi Rexha,
Mauro Dragoni,
Roman Kern

Abstract

Digital Libraries benefit from the use of text classification strategies since they are enablers for performing many document management tasks like Information Retrieval. The effectiveness of such classification strategies depends on the amount of available data and the classifier used. The former leads to the design of data augmentation solutions where new samples are generated into small datasets based on the semantic similarity between existing samples and concepts defined within external linguistic resources. The latter relates to the capability of finding, which is the best learning principle to adopt for designing an effective classification strategy suitable for the problem. In this work, we propose a neural-based architecture thought for addressing the text classification problem on small datasets. Our architecture is based on BERT equipped with one further layer using the sigmoid function. The hypothesis we want to verify is that by using embeddings learned by a BERT-based architecture, one can perform effective classification on small datasets without the use of data augmentation strategies. We observed improvements up to 14% in the accuracy and up to 23% in the f-score with respect to baseline classifiers exploiting data augmentation.

5.1 Introduction

The new spring of Artificial Intelligence (AI) opened the opportunity of designing and implementing textual classification strategies in many research fields that can benefit, in particular, by the integration of neural-based solutions into existing classifiers. On the one hand, such an integration demonstrated to be suitable for improving the overall effectiveness of existing textual classifiers. On the other hand, researchers started to deal with the problem of missing data. Indeed, many domains suffer from the lack of datasets mainly, due to the costs of manual labeling.

One of the literature's proposed techniques for addressing this challenge is based on *data augmentation* (Elekes et al., 2019). Briefly, it consists of expanding original datasets with new samples created by applying proper similarity metrics by taking into account the features used for building the classification model. Data augmentation is used in many tasks ranging from the Natural Language Processing (NLP) ones (e.g., word sense disambiguation, sentiment analysis, etc.) to image and video recognition. Within the NLP domain, conventional techniques are built upon the use of distributional semantic strategies applied for selecting similar terminology to include in the automatically generated samples.

While data augmentation solutions demonstrated to be effective, the use of artificial samples may have detrimental effects on the overall effectiveness of the generated classification model. This problem mainly occurs since a domain expert does not supervise the new samples. Hence, they can introduce errors in the generated models. This aspect has been demonstrated in the literature by considering, for example, query expansion strategies (Cronen-Townsend, Zhou, and Croft, 2004; Abdelali, Cowie, and Soliman, 2007).

In this paper, we propose a neural-based architecture designed for addressing the challenge described above. Such an architecture has been thought for learning effective models by working on small datasets. With this work, we want to answer the following research questions:

1. Is it possible to design a neural-based architecture able to build effective models from small datasets, that outperforms state of the art data augmented techniques?

2. By using such an architecture, would the use of data augmentation techniques have a detrimental effect on the overall effectiveness or, at least, to have not significant improvements?

Concerning the first research question, the proposed approach has been validated, on four datasets publicly available, by comparing obtained results three baselines models trained by exploiting state of the art augmentation techniques for digital libraries. In particular, in this work, we used as baselines the classifiers trained with augmented datasets presented in Elekes et al., 2019. While answering the second research question, we applied three different data augmentation techniques on all datasets and compared the results with respect to the models trained on the original datasets.

The remainder of the paper is structured as follows. Section 5.2 surveys the most recent advances about strategies on building classifiers for small textual datasets and on the main data augmentation techniques by highlighting their limits. In Section 5.3, we present our neural-based architecture and we introduce the state of the art data augmentation strategies we implemented for demonstrating how data augmentation may have a detrimental effect when applied to classification strategies that are already effective. Section 5.4 provides the evaluation we performed on four state of the art datasets and compares the obtained results with respect to the recent literature in data augmentation. Then, in Section 5.5, we discuss the results obtained by our neural-based architecture when trained on augmented datasets. Finally, Section 5.6 concludes the paper.

5.2 Related Work

This work considers the case of textual classification for small datasets. Given the two research questions introduced in Section 5.1, we surveyed the main recent contributions related to text classification on small datasets and about the impact of the most relevant data augmentation techniques on textual datasets.

Small Dataset Textual Classification One of the early works on small datasets is the hierarchical document classification introduced in Toutanova et al., 2001. This work uses a Bayesian approach to classify documents in a hierarchy of topics. Differently from previous attempts, the inner nodes of the classifier "update" their class conditional probability of each word. Thus, obtaining a differentiation of terms in the hierarchy according to their level of generality/specificity.

In a less complicated scenario, Shridhar et al., 2019 focuses on the task of Intent Classification. The main issue with conventional methods is that they do not consider spelling errors and out of vocabulary words. The paper proposes the use of Semantic Hashing as embeddings. Precisely, in the first step, the sentence is split in its words, and secondly, each word is divided into character 3-grams. These 3-grams are then used as features, weighted with a Term Frequency - Inverse Document Frequency (TF-IDF), normalized, and passed to different classifiers.

A topic modeling approach is used in Clarizia et al., 2011 for classifying small datasets. The method creates a graph where words are connected to their topics with edges representing the probability of the word being part of it. Topics are then connected via a probability model. Hence, a document is described as a probability model over the topics graph, where these probabilities represent the features. The authors achieve similar results to the state of the art of the time, with around 1% of the data.

Kim et al., 2019 tackles the case of insufficient training data on a large number of categories. A novel architecture for multi-task learning is proposed for such a scenario. A convolutional neural network is created and trained small- and large-scale classification tasks that are considered related. Their shared features are then combined for improving the missing of the instances in the training set. Moreover, they use over-sampling in the skewed dataset, which duplicates the under-represented classes' instances until the dataset is balanced.

The goal of Kou et al., 2020 is to identify the best feature selection methods (i.e., information gain, Gini index, etc.) for classification tasks in small datasets. The authors also argue that it is necessary to also value other evaluations rather than accuracy, like stability, efficiency, etc. So they propose an evaluation based on multiple criteria decision making(MCDM):

TOPSIS, VIKOR, GRA, Weighted sum method(WSM), and PROMOTHEE. Ten datasets are picked for choosing the best feature selector, with nine measures for binary- and seven for multiclass- classification.

Data Augmentation of Textual Dataset To augment the textual dataset, Lu et al., 2006 uses a method based on Latent Dirichlet Allocation (LDA). The output of the LDA is employed as enrichment in the form of embeddings of documents to words. LDA is also used in Abulaish and Sah, 2019 as a keyword extractor, where the authors extract the top keywords for each class of the sentiment task. The keywords of each class are tested whether they are contained in the 3-grams of the instances of the class. In such a case, the instance owning the 3-gram is enriched with the same 3 grams, thus boosting its importance.

A different approach, which is based on the syntactic modification of the text, is presented in Wei and Zou, 2019. This work proposes a system called EDA (Easy Data Augmentation), which implements four different methods for enriching the dataset. The first enrichment substitutes "n" random words of the text with their synonyms, while the second inserts the synonym of a randomly selected word in the instance. Random swap and deletion are the two other methods proposed to change the text. The former approach swaps two words, while the latter deletes another one with a random selection. The output of these methods is a new instance that is added to the dataset.

The study in Yu et al., 2019 proposes a new system called Hierarchical Data Augmentation (HDA) and compares it to EDA. The authors stack two levels of bidirectional Gated Recurrent Units (GRU), each supported by an attention layer: the first layer used for word attention and the second used for sentence attention. In the first step, the training set is passed to the network responsible for the classification. The attention layers of the network are then used to create a new training document with the most important (in terms of attention) words and sentences.

In Rizos, Hemker, and Schuller, 2019, the authors propose three techniques for increasing the number of instances for imbalanced classes. In the first technique, words are substituted with synonyms with the same part of speech. The synonyms are identified via cosine similarity on pre-trained

word embeddings. The second technique was inspired by the variation of the sentence length in the sample batches. This technique is more of an engineering step, as it adds to the system perturbed sentences. So, instead of only left padding in batches with 0 values, the idea here is to shift the sentence within the sample boundaries. This step is needed as the last augmented technique uses a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) network augmented with a Fully Connected Neural Networks (FCNN). The final output is fed to a softmax neuron that outputs the corresponding class of the input. Results between data augmentation and the original data do not show a significant benefit. The only benefit visible from the paper is the increased recall of hate speech (smaller class) on the short text.

Enrichment on a tiny spam dataset for short-text messages is presented in Lochter et al., 2018. Three steps are applied to create new instances. The first is the normalization of the text (by removing grammatical errors) via an English dictionary. Next, a semantic indexing technique is used to get synonyms for words, with the synonyms filtered with a concept disambiguation tool. Thus, a new sample is generated using one synonym from each set at each step.

Data enrichment also applies in other languages than English. A strategy of augmentation for Chinese is presented in Sun, He, and Quan, 2017 with two different enrichment strategies: 1) at the word level and 2) at the phrase level. At the word level, synonyms are exchanged, and random meaningless words added as noise. While on the phrase level, the substitution is done at the adverbial phrases by word2vec and thesaurus. The system assesses the results in sentiment analysis for the publicly available hotel online evaluation dataset.

In Veliz, Clercq, and Hoste, 2019 the authors explore how to overcome this data bottleneck for Dutch, a low-resource language. The goal of the paper is to normalize the text as defined in Schulz et al., 2016, going from noisy data to standard text. Three manually annotated datasets are provided: Tweets, Message Board Posts, and Text Messages (SMS). The enrichment is performed with a distance-based substitution on word embeddings. The unnormalized enriched instances are then passed to Sequence-to-Sequence

(seq2seq) classification (encoder-decoder architecture) with the normalized instances as the target.

A work that goes in a similar direction to ours is Wu et al., 2019. In this work, the authors finetune the BERT (explained in detail in the next section) masked terms, with specific terms per class. It means that instead of predicting only the most probable word, they also take into account the class in which it occurs. By using this enrichment, the authors show a tiny improvement compared to the use of simple BERT.

5.3 Method

In this section, we present the proposed neural-based architecture. We start to introduce in subsection 5.3.1 the word embedding we adopted for the setup of our classifier. Then, in subsection 5.3.2, we describe how these embeddings have been exploited and how we configured our neural architecture. Finally, subsection 5.3.3, introduces the state of the art data augmentation strategies we implemented for validating the second research question presented in Section 5.1 concerning the detrimental effects of augmented datasets when adopted on already-effective classifiers.

5.3.1 Word Embeddings

Word embeddings were introduced to avoid the curse of dimensionality for NLP tasks such as Text Classification, Relation Extraction, etc. In early systems, for each word in the training dataset, a single multidimensional representation is learned. Among the systems of this category, which we call non-contextual, one can find Word2Vec (Mikolov et al., 2013), GloVe (Pennington, Socher, and Manning, 2014), fastText (Joulin et al., 2016), etc.. The learned embeddings from these systems are very powerful and have proven to work quite well in different NLP tasks, but sometimes lack in the contextual representation of the word itself. For example, the word bank can have different senses. It could refer to a riverbank or a financial institution. This ambiguity may mislead systems built for NLP tasks and trained on

these embeddings. As a solution, a new family of contextual word embedding systems has been proposed. These systems do not learn a single word embedding but one depending on its context. Such systems are often used for transfer learning, where pre-trained models are made available to NLP practitioners. Furthermore, the latest ones allow fine-tuning of their weights in the network for cascading tasks. This way, the distributions of the data in the embeddings can better represent the one at hand.

One of the best-known systems from this type of embeddings is BERT (Bidirectional Encoder Representations from Transformers from Devlin et al., 2019). BERT is based on the well-known Transformer (Vaswani et al., 2017) architecture (encoder-decoder) and uses only the encoder part of it. The main advantage of such architecture is exploiting the bidirectional structure of the attention mechanism. To train BERT, the authors propose the following two steps:

1. Mask tokens in a sentence with a 15% probability and try to predicting them (a modified version of Language Models called Masked Language Model).
2. Predict whether two sentences are written one after each other.

Thus, initially, BERT is trained (i) with generic data for identifying missing tokens, and then (ii) it is further fine-tuned for tasks like question answering. We use BERT for transfer learning to improve the performance (i.e., accuracy) of classification tasks in small datasets. Intuitively, BERT learns some distribution over the generic corpora and adapts it to the words in the current dataset. This type of architecture gives a lot of advantages for organizations which do not have the computational power to train on a huge dataset, resulting in a state of the art performance on cascading tasks. In this paper, we use BERT two-fold, as an input for a classification algorithm and for enriching small datasets. A detailed description of its usage can be found in the next subsections.

5.3.2 BERT-Based Classifier

To improve the current baselines of classification tasks on small datasets, we use a BERT based classifier. As previously mentioned, BERT is the encoder

part of a Transformer architecture. This part is composed of 12 stacked (encoder) blocks [1] with each of them consisting of 3 components:

- **Multi-head attention** is the first component of an encoder block. It uses a set of parallel self-attention networks to learn the closeness relation of the word itself (hence "self"-attention) to embeddings of the other words.
- The **Feedforward** component consists of two linear layers with a ReLU activation function connecting them. This part returns an embedding representation for each word received from the output of the previous component.
- Two **layer normalization** (Jia, Kiros, and Ba, 2019) components, with the first located between the two previously described components and the second located before the output of the block.

The position embedding is another peculiarity of BERT. BERT uses an embedding layer to encode it and updates it during training instead of having the position of the word as raw input. We combine such architecture with a sigmoid function as the last layer with a dropout rate of 0.1 to avoid overfitting. Figure 5.1 illustrates the classification system. The output of the network is presented by:

$$h(x) = \sigma(\theta^T x(\gamma) + b) = \frac{1}{1 + e^{-(\theta^T x(\gamma) + b)}}$$

where $x(\gamma)$ are the outputs of BERT for the text at hand, γ represent the weights of BERT and θ are the learned weights and b is the bias term of the output neuron.

The expected output to minimize is the function that predicts the target labels in the best manner. These are the labels of each short text with the cost function of the network defined as following:

$$J = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log(h^{(i)}) + (1 - y^{(i)}) \log(1 - h^{(i)})$$

[1] The number of layers is 6 in the pre-trained model that we chose to use called *BERT_Base, Uncased*

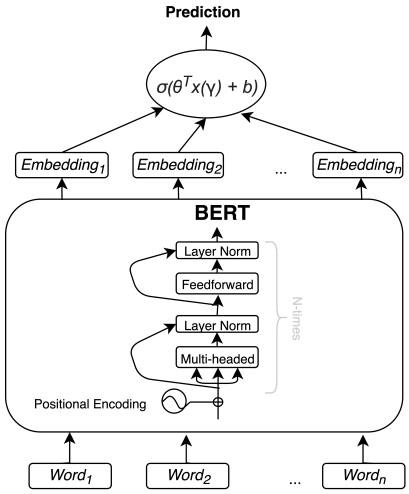


Figure 5.1: The architecture of our classifier. A BERT architecture with a sigmoid function on top for a two class classification.

Where m is the number of instances in the training set, $y^{(i)}$ is the label of the $i-th$ instance, and $h^{(i)}$ is the output of the sigmoid layer for that instance. The goal of the algorithm is to find the γ, θ, and b that would minimize the cost function: $args\_min_{\gamma,\theta,b}(J)$. As can be noticed, the goal of the algorithm isn't only to update the weights of the output neuron. During training, the weights of BERT are updated to make the network adapt to the distribution of the training set. We use this classifier as opposed to previously proposed baselines and for assessing data augmentation strategies. In our configurations, we use batches of 8 instances and train the dataset for 10 epochs.

5.3.3 Enriching the Dataset

To improve the performance (i.e., accuracy) of classification tasks on small datasets, here we try different strategies for enriching. Traditionally, to augment textual datasets, new samples with substituted words (usually synonyms) are introduced in the data. We use a similar approach for our experiments, but instead of using traditional external resources or non-contextual word embeddings, we exploit BERT's ability as a Masked Language Model

(MLM). We predict missing words by masking them in the training set and substituting them with their most prominent candidates. More precisely, given a text $t = w_1 w_2 \dots w_n$, with the word $w_m = [MASK]$, we are interested in extracting $top\_args(P(BERT(w_m|t)))$, where top_args returns the words with the highest BERT probability. To achieve good results, we need to pick and define the following parameters carefully:

- Which tokens to substitute?
- How many substitutes to use?
- What kind of $top\_args()$ function to use?

To identify the word to substitute in each text, we opted for a Term Frequency - Inverse Document Frequency (TF-IDF) weighting scheme. The idea behind such selection is that terms that occur in fewer documents are more discriminating and thus more important. For each word w_i in the text $t_j = w_1 w_2 \dots w_n$ in the training set, we calculate:

$$tfidf_{i,j} = tf_{i,j} \cdot idf_i$$

where $tf_{i,j}$ represents the term frequency of word i in the document j defined as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ represents the number of times that the word i occurs in the text j and idf represents the inverse document frequency of word i calculated as:

$$idf_i = \log \frac{|t|}{|t_j : w_i \in t_j|}$$

Next, we select the most prominent candidates for word substitution ($top\_args()$) with three different configurations: 20%, 40% and 60% of the top $tfidf$ weights. Also, we decide to use 4 possible values for the maximum amount of substitutions: [3, 5, 7, 9]. This means that we select the top 3, 5, 7 or 9 closest terms with our candidate word returned from BERT.

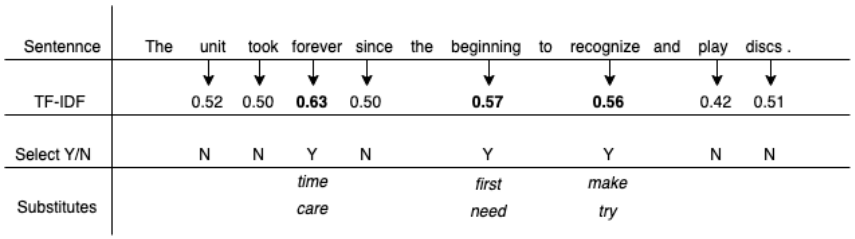


Figure 5.2: Example of selection of words to substitute for enrichment. First a selection based on TF-IDF is made. Later, for each selected word, BERT is used to find the substitutes.

Furthermore, we do not consider stopwords for this process. Figure 5.2 illustrates an example of the enrichment technique described above.

In this work, we try three different filtering strategies for enriching the text in small datasets. Algorithm 1 presents the generic logic of these filtering strategies. Each of them differs from each other in the **filter()** function (bolded in the Algorithm 1). In the **first filtering strategy** (raw strategy), *the Naive BERT*, the filter function does not remove any of the substitution candidates.

One of the questions that arises is about the amount of masked tokens we need to extract for replacements. We use 4 different values to get the maximum amount of replacement tokens: 3, 5, 7, 9.

Once we have identified the technique to use for enriching the dataset, we have to decide which words to substitute. At the beginning we use a TF-IDF weighting scheme for each word in the candidate text. Then we select 3 different percentages from the selected words: 20%; 40%; 60% of the best candidates.

By analyzing the enriched text, we discovered that sometimes out-of-context words are replaced. Other than that, as expected, BERT returns both synonyms and antonyms. Obviously, for some tasks (i.e., sentiment analysis), antonyms substitutions are counter-productive. Nevertheless, for others (i.e., subjectivity), antonyms substitutions are very beneficial. They would make the new instance semantically correct. To try only synonyms and remove the out-of-context substitutes, we propose the **second filtering strategy** (synonym strategy) that replaces tokens only with synonyms returned

Algorithm 1 The algorithm for enriching the training set.

Require: ts, the text of the training set

for each $t \in ts$ **do**
  $tfidfs \leftarrow []$
  for each $w \in t$ **do**
    if $w \notin stopwords$ **then**
      $weight_w \leftarrow tfidf(w)$
      $tfidfs.append(w, weight_w)$
    end if
  end for
  $candidate \leftarrow best - weighted(tfidfs)$
  for each $c \in candidates$ **do**
    $substitutes = get\_substitutes(c)$
    $final\_substitutes \leftarrow$ **filter(substitutes)**
    for each $sub \in final\_substitutes$ **do**
      $new\_text \leftarrow exchange(t, c, sub)$
      $ts.append(new\_text)$ ▷ Append candidate
    end for
  end for
end for

by BERT. To do so, we extract all synonyms of the candidate word from WordNet and accept them if and only if they have the same stem as the replacement returned by BERT. Hence, we only have synonym replacements from a contextual language model. Thus the new instance is "natural". The difference of this filtering strategy with the previous one is on the **filter()** method of the Algorithm 1. Algorithm 2 presents the logic of acceptance or rejection of the candidate substitutions.

Algorithm 2 Whether to accept or reject a possible substitute with WordNet.

Require: c, sub, candidate to substitute and possible substitute
1: $syns \leftarrow wordnet.synset(c)$
2: $stemsyns \leftarrow stem(syns)$
3: **if** $stem(substitute) \in stemsyns$ **then**
4: accept ▷ Only accept if at least one stem matches
5: **else**
6: reject
7: **end if**

One of the possible drawbacks of the "synonym" strategy is that particular synonyms substitution might be part of the other class. By substituting words that occur in the other class (or the candidates of the other class), we might introduce confusion to the classifier. To avoid such behavior, we propose a **third filtering strategy** (pure strategy) that further filters the possible synonyms (intersection between WordNet and BERT) by rejecting substitution candidates that are part of the words in the other class and/or even possible substitution of the other class. Our goal in this filtering strategy is to enrich the training only with "more pure" information.

5.4 Validation of The Proposed Neural Architecture

In this section, we present the evaluation performed on four short-text datasets benchmarked in numerous NLP studies: Customer Reviews (CR),

Dataset	Documents	# of Words	Average Doc. Length	Positive Docs	Negative Docs
CR	3,772	6,596	20	2,406	1,366
MPQA	10,624	6,298	3	3,316	7,308
Rt10k	10,662	20,621	21	5,331	5,331
Subj	10,000	23,187	24	5,000	5,000

Table 5.1: Statistics of the dataset used for the evaluation. The first two columns contain the number of documents and the different words contained in each dataset, respectively. In the third column the average number of words composing each sample of the dataset is depicted. The last two columns show the number of positive and negative samples contained in each dataset, respectively.

MPQA, Short Movie Reviews (Rt10k) and Subjectivity (Subj)[2]. A summary of additional information about the employed datasets can be seen in Table 5.1. All these datasets are cases of binary classification.

We subdivide each dataset as follows: the test sets consist of 1000 samples held out from each dataset for later testing. To show the effectiveness of our method with respect to the techniques presented in Elekes et al., 2019. We applied the designed neural architecture to different training-set sizes: 500, 1000, 1500, 8500 for MPQA, Rt10k and Subj and 500, 1000, 1500, 2600 for CR. For each size, we sampled five training sets randomly, using stratified sampling. For each of these sets, a 10-fold cross-validation is performed to find the best parameter combination, i.e., the combination that yields the highest average accuracy over all folds. The classifier was then trained on the same dataset used for the cross-validation and tested on the held-out test set (five times for each sample size and classifier combination).

As baselines, we refer to three approaches used in a recent state of the art evaluation as discussed in Elekes et al., 2019 namely the Multinomial Naive Bayes (MNB), the Naive Bayes Support Vector Machine (NBSVM), and a Recursive Auto-Encoder (RAE). The MNB and NBSVM classifiers have been parametrized, as discussed in Wang and Manning, 2012. While the word embeddings of the RAE classifier have been initialized as suggested by the authors [3] Socher et al., 2011. For all the baselines and the proposed approach

[2]All datasets are available at https://github.com/sidaw/nbsvm.

[3]A MATLAB implementation of the classifier is available at http://www.socher.org.

Dataset	BERT	MNB	NBSVM	RAE
CR 500	**0.906**	0.743	0.759	0.758
CR 1000	**0.908**	0.779	0.779	0.797
CR 1500	**0.912**	0.782	0.787	0.813
CR 2600	**0.916**	0.810	0.808	0.806
MPQA 500	**0.870**	0.778	0.769	0.775
MPQA 1000	**0.882**	0.804	0.804	0.834
MPQA 1500	**0.876**	0.824	0.812	0.832
MPQA 8500	**0.902**	0.888	0.877	0.879
Rt10k 500	**0.828**	0.674	0.646	0.671
Rt10k 1000	**0.844**	0.683	0.680	0.714
Rt10k 1500	**0.852**	0.705	0.715	0.695
Rt10k 8500	**0.880**	0.756	0.778	0.792
Subj 500	**0.940**	0.861	0.855	0.893
Subj 1000	**0.946**	0.880	0.869	0.901
Subj 1500	**0.960**	0.897	0.881	0.911
Subj 8500	**0.966**	0.922	0.916	0.957

Table 5.2: Summary of the accuracies observed for the proposed approach and for the three baselines.

we reported the accuracies in Table 5.2 and the precision, recall, and f-score in Table 5.3.

Results reported in Table 5.2 related to the accuracies show how the proposed approach outperforms all the baselines. The improvements range from around 5% for the MPQA and Subj datasets to approximately 10% for the CR and Rt10k ones. By analyzing these results from the dataset statistics perspective, we did not find any specific correlation. However, we observed that if we rank the datasets by the effectiveness of each classifier, we can notice that such a rank is the same for all classifiers. Such results suggest that even if the proposed strategy significantly improves the classification capabilities, there might be a subset of samples that are quite challenging due to their particular structure. In the future, we will focus on the manual analysis of such examples. We intend to extract them from the error analytics of each classifier to understand which are the reasons that led to the errors.

Besides the observation of the accuracies obtained by each classifier, we also computed the precision, recall, and f-score as reported in Table 5.3.

Dataset	BERT			MNB			NBSVM			RAE		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
CR 500	**0.906**	**0.906**	**0.906**	0.740	0.448	0.558	0.698	0.588	0.639	0.740	0.428	0.492
CR 1000	**0.908**	**0.908**	**0.908**	0.747	0.588	0.658	0.717	0.630	0.671	0.747	0.513	0.604
CR 1500	**0.912**	**0.912**	**0.912**	0.729	0.633	0.678	0.708	0.702	0.705	0.729	0.653	0.682
CR 2600	**0.916**	**0.916**	**0.916**	0.737	0.666	0.700	0.736	0.732	0.734	0.737	0.728	0.738
MPQA 500	**0.872**	0.870	0.870	0.824	0.877	0.850	0.852	0.820	0.836	0.784	**0.941**	**0.872**
MPQA 1000	**0.882**	0.882	0.882	0.838	0.901	0.868	0.878	0.844	0.860	0.818	**0.947**	**0.899**
MPQA 1500	0.878	0.876	0.876	0.844	0.925	0.883	**0.888**	0.844	0.865	0.834	**0.965**	**0.911**
MPQA 8500	0.902	0.902	0.902	0.916	0.929	**0.922**	**0.937**	0.888	0.912	0.876	**0.930**	0.918
Rt10k 500	**0.828**	**0.828**	**0.828**	0.672	0.601	0.645	0.679	0.595	0.632	0.672	0.690	0.684
Rt10k 1000	**0.844**	**0.844**	**0.844**	0.705	0.654	0.678	0.698	0.659	0.678	0.705	0.630	0.666
Rt10k 1500	**0.854**	**0.852**	**0.852**	0.731	0.701	0.715	0.722	0.718	0.720	0.731	0.652	0.694
Rt10k 8500	**0.882**	**0.880**	**0.880**	0.776	0.736	0.757	0.788	0.748	0.767	0.776	0.821	0.801
Subj 500	**0.944**	0.940	**0.940**	0.878	0.845	0.861	0.867	0.845	0.856	0.878	**0.956**	0.908
Subj 1000	**0.950**	0.946	**0.946**	0.900	0.861	0.880	0.882	0.859	0.870	0.900	**0.979**	0.933
Subj 1500	**0.960**	**0.960**	**0.960**	0.913	0.883	0.898	0.891	0.865	0.874	0.913	0.937	0.923
Subj 8500	**0.968**	**0.966**	**0.966**	0.942	0.900	0.920	0.923	0.912	0.917	0.949	0.930	0.938

Table 5.3: Summary of the Precision, Recall, and F-score observed for the proposed approach and for the three baselines.

Here, we can notice a more interesting scenario. Indeed, while for the accuracies, our approach outperforms all baselines for each dataset and training size, the situation is different by taking into account the precision, recall, and f-score. The reader may notice that for the MPQA dataset, the RAE classifier obtained the best f-score due to the very high recall values. Similarly, it also occurs on the Subj dataset, where two out of four training-sized configurations, the RAE classifier outperforms our approach. The analysis of the error matrices highlighted how the RAE classifier performed very well in detecting false-negative samples, while it has poor performance on detecting the false positive ones. This behavior causes the low precision values in favor of the recall ones. Except for the MPQA dataset, our approach outperforms all the baselines with a delta of more than 15%.

Finally, we performed an error analysis of the obtained results to have a deeper understanding of the classifier's behavior. In particular, we want to observe if the classifier is biased for specific classes in unbalanced datasets. We illustrate the colored confusion matrices in Figure 5.3. We don't show the real numbers there as the colors demonstrate the point we want to make. Concerning the balanced datasets (Rt10k and Subj), we can notice how the classifier encounters the same error rate for both classes (similar color). Instead, for the unbalanced datasets (Cust and MPQA), we can appreciate that the higher error rate has been observed in the most represented classes. This means that the classifier has good generalization capabilities since its performance is not affected by the number of samples contained in each class.

In this section, we demonstrated how neural architecture designed to address the challenge of managing small datasets could result in significant improvements in data augmentation solutions integrated into different classifiers. In the next section, we show how the integration of data augmentation strategies is not particularly useful when the neural architecture has already been designed for addressing a specific task.

Dataset	No Enrichment	Raw	Synonym	Pure
CR 500	**0.906**	0.886	0.885	0.880
CR 1000	**0.908**	0.891	0.892	0.892
CR 1500	**0.912**	0.887	0.893	0.902
CR 2600	**0.916**	0.892	0.905	0.899
MPQA 500	0.870	**0.893**	0.868	0.871
MPQA 1000	0.882	**0.899**	0.880	0.889
MPQA 1500	0.876	**0.901**	0.883	0.891
MPQA 8500	**0.902**	0.899	0.877	0.901
Rt10k 500	**0.828**	0.814	0.816	0.781
Rt10k 1000	**0.844**	0.814	0.834	0.799
Rt10k 1500	**0.852**	0.839	0.845	0.812
Rt10k 8500	**0.880**	0.842	0.846	0.814
Subj 500	0.940	0.928	**0.949**	0.946
Subj 1000	0.946	0.946	0.952	**0.957**
Subj 1500	0.960	0.959	**0.968**	0.953
Subj 8500	0.966	0.963	**0.967**	0.951

Table 5.4: Summary of the accuracies observed for the classifier comparing the three proposed setups for enrichment and the original dataset.

5.5 Effects of Data Augmentation

In this section, we present the results observed by applying the three augmentation techniques described in Section 5.3.3, and we discuss if the use of such techniques helps in improving the overall effectiveness of the classifier or not.

Table 5.4 shows the accuracies obtained by our approach (second column) for the three strategies (from third to fifth columns). We can appreciate how our approach outperforms the models trained with the augmented datasets in 9 out of 16 cases. For what concerns the Subj dataset, our method is less effective than the augmented models. However, the gap is minimal and is not statistically significant.

On the contrary, for the first three configurations of the MPQA dataset, the *raw* augmented strategy significantly outperforms the others.

The same result if reflected in the precision, recall, and f-score reported in Table 5.5. Indeed, also here we can notice how, besides the first three

Dataset	No Enrichment			Raw			Synonym			Pure		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
CR 500	**0.906**	**0.906**	**0.906**	0.885	0.886	0.885	0.885	0.885	0.885	0.881	0.880	0.880
CR 1000	**0.908**	**0.908**	**0.908**	0.891	0.891	0.889	0.893	0.892	0.892	0.893	0.892	0.890
CR 1500	**0.912**	**0.912**	**0.912**	0.886	0.887	0.886	0.894	0.914	0.904	0.902	0.902	0.900
CR 2600	**0.916**	**0.916**	**0.916**	0.892	0.895	0.893	0.905	0.906	0.905	0.898	0.900	0.899
MPQA 500	0.872	0.870	0.870	**0.891**	**0.893**	**0.892**	0.873	0.868	0.869	0.873	0.874	0.873
MPQA 1000	0.882	0.882	0.882	**0.898**	**0.899**	**0.896**	0.879	0.880	0.877	0.887	0.889	0.887
MPQA 1500	0.878	0.876	0.876	**0.900**	**0.901**	**0.900**	0.883	0.887	0.885	0.889	0.891	0.890
MPQA 8500	**0.902**	0.902	0.902	0.899	0.904	0.901	0.877	0.880	0.878	0.901	**0.905**	**0.903**
Rt10k 500	**0.828**	**0.828**	**0.828**	0.814	0.814	0.8139	0.823	0.816	0.814	0.781	0.782	0.781
Rt10k 1000	**0.844**	**0.844**	**0.844**	0.827	0.814	0.811	0.834	0.834	0.823	0.799	0.796	0.797
Rt10k 1500	**0.854**	**0.852**	**0.852**	0.839	0.839	0.839	0.845	0.845	0.845	0.812	0.811	0.812
Rt10k 8500	**0.882**	**0.880**	**0.880**	0.841	0.842	0.842	0.846	0.846	0.846	0.813	0.814	0.814
Subj 500	**0.944**	0.940	**0.940**	0.928	0.928	0.928	0.949	0.949	0.949	0.946	0.946	0.946
Subj 1000	**0.950**	0.946	**0.946**	0.946	0.946	0.946	0.952	0.952	0.952	**0.957**	**0.957**	**0.957**
Subj 1500	0.960	0.960	0.960	0.959	0.959	0.959	**0.967**	**0.967**	**0.967**	0.913	0.937	0.923
Subj 8500	**0.968**	0.966	0.966	0.964	0.963	0.963	0.966	**0.968**	**0.967**	0.951	0.951	0.951

Table 5.5: Summary of the Precision, Recall, and F-score observed for data data that aren't augmented and the three augmeted setups.

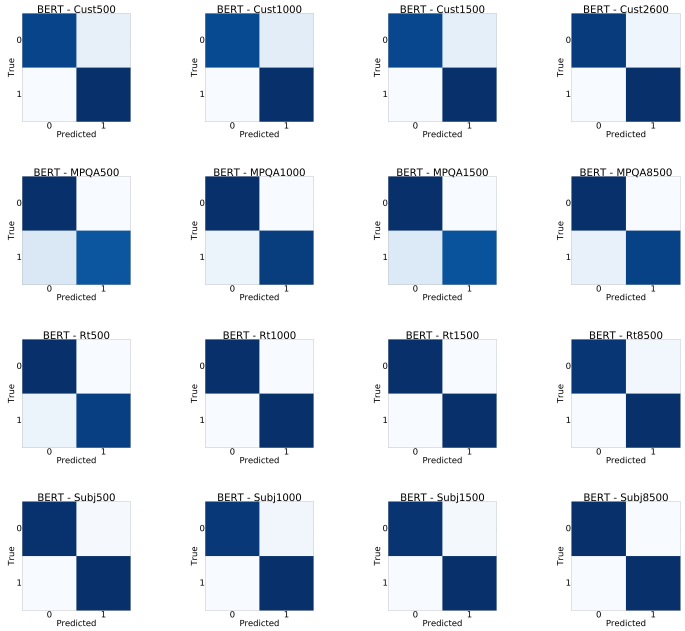


Figure 5.3: Confusion matrices computed on the results obtained by the proposed classifier. A darker color indicates a larger number of instances. We use only colors to show how the classifier performs approximately for each class.

configurations of the MPQA dataset, the proposed approach outperforms the models trained with the augmented datasets.

These results allow us to answer the second research question since the implementation of state of the art data augmentation techniques, in general, did not significantly improve the classifier. However, it would be necessary to perform a more in-depth analysis of the classifier errors to find which are the reasons for such a detrimental effect. Finally, for having a complete picture of the classifier behavior, the system should be equipped with an explainable model providing further details about why a specific sample is classified in a particular way. All these aspects will be part of future work.

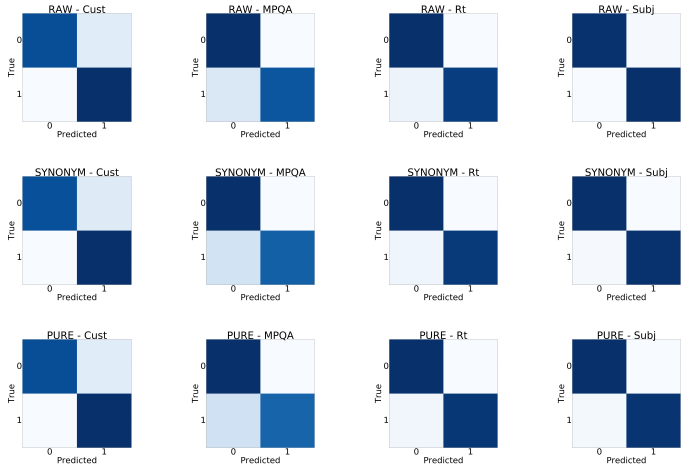


Figure 5.4: Confusion matrices computed on the results obtained by the proposed classifier trained on the dataset enriched with the three described data augmentation strategies. A darker color indicates a larger number of instances. We use only colors to show how the classifier performs approximately for each class.

5.6 Conclusion

In this paper, we discussed how the use of a neural-based architecture designed for addressing the task of text classification on small datasets outperforms models trained with augmented datasets. We provided two research questions which we positively answered.

The first research question was related to the comparison between the results obtained by the proposed strategies and the ones obtained by other three baseline systems. By observing the results, reported in Section 5.4, we can state that the approach presented in this paper is suitable with respect to the use of classifier trained with augmented dataset.

The second research question, instead, was related to the possible detrimental effects that the use of augmented datasets can have on classifiers that have been already tuned for working on small datasets. Results reported in Section 5.5 confirmed this hypothesis since the effectiveness of the three

state of the art strategies we implemented for augmenting the dataset did not improve the effectiveness of the classifier significantly.

As a general note, we believe that our system would perform in a similar way for multi-class classification. The only difference to make to the architecture is in the last layer. Indeed, instead of having only one output with a sigmoid activation function, we have to add a softmax activation function.

Future work will focus on performing a deeper analysis of the errors we reported and discussed in Section 5.4 and 5.5. With the aim of inferring if there are common characteristics among the samples classified wrongly, we intend to design an explainable strategy able to extract such characteristics directly from the trained model.

Bibliography

Abdelali, Ahmed, Jim Cowie, and Hamdy S. Soliman (2007). "Improving query precision using semantic expansion." In: *Inf. Process. Manage.* 43.3, pp. 705–716. DOI: 10.1016/j.ipm.2006.06.007. URL: https://doi.org/10.1016/j.ipm.2006.06.007 (cit. on p. 98).

Abulaish, Muhammad and Amit Kumar Sah (2019). "A Text Data Augmentation Approach for Improving the Performance of CNN." In: *11th International Conference on Communication Systems & Networks, COMSNETS 2019, Bengaluru, India, January 7-11, 2019*. IEEE, pp. 625–630. DOI: 10.1109/COMSNETS.2019.8711054. URL: https://doi.org/10.1109/COMSNETS.2019.8711054 (cit. on p. 101).

Clarizia, Fabio et al. (2011). "A new text classification technique using small training sets." In: *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011*. Ed. by Sebastián Ventura et al. IEEE, pp. 1038–1043. DOI: 10.1109/ISDA.2011.6121795. URL: https://doi.org/10.1109/ISDA.2011.6121795 (cit. on p. 100).

Cronen-Townsend, Stephen, Yun Zhou, and W. Bruce Croft (2004). "A framework for selective query expansion." In: *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*. Ed. by David A. Grossman et al. ACM, pp. 236–237. DOI: 10.1145/1031171.1031220. URL: https://doi.org/10.1145/1031171.1031220 (cit. on p. 98).

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423 (cit. on p. 104).

Elekes, Ábel et al. (2019). "Learning from Few Samples: Lexical Substitution with Word Embeddings for Short Text Classification." In: *19th ACM/IEEE Joint Conference on Digital Libraries, JCDL 2019, Champaign, IL, USA, June 2-6, 2019*. Ed. by Maria Bonn et al. IEEE, pp. 111–119. DOI: 10.1109/JCDL.2019.00025. URL: https://doi.org/10.1109/JCDL.2019.00025 (cit. on pp. 98, 99, 111).

Jia, Sheng, Jamie Kiros, and Jimmy Ba (2019). "DOM-Q-NET: Grounded RL on Structured Language." In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. URL: https://openreview.net/forum?id=HJgd1nAqFX (cit. on p. 105).

Joulin, Armand et al. (2016). "Bag of Tricks for Efficient Text Classification." In: *arXiv preprint arXiv:1607.01759* (cit. on p. 103).

Kim, Kang-Min et al. (2019). "From Small-scale to Large-scale Text Classification." In: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. Ed. by Ling Liu et al. ACM, pp. 853–862. DOI: 10.1145/3308558.3313563. URL: https://doi.org/10.1145/3308558.3313563 (cit. on p. 100).

Kou, Gang et al. (2020). "Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods." In: *Appl. Soft Comput.* 86. DOI: 10.1016/j.asoc.2019.105836. URL: https://doi.org/10.1016/j.asoc.2019.105836 (cit. on p. 100).

Lochter, Johannes V. et al. (2018). "Semantic Indexing-Based Data Augmentation for Filtering Undesired Short Text Messages." In: *17th IEEE International Conference on Machine Learning and Applications, ICMLA*

2018, Orlando, FL, USA, December 17-20, 2018. Ed. by M. Arif Wani et al. IEEE, pp. 1034–1039. DOI: 10.1109/ICMLA.2018.00169. URL: https://doi.org/10.1109/ICMLA.2018.00169 (cit. on p. 102).

Lu, Xinghua et al. (2006). "Research Paper: Enhancing Text Categorization with Semantic-enriched Representation and Training Data Augmentation." In: *JAMIA* 13.5, pp. 526–535. DOI: 10.1197/jamia.M2051. URL: https://doi.org/10.1197/jamia.M2051 (cit. on p. 101).

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781* (cit. on p. 103).

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "Glove: Global vectors for word representation." In: *In EMNLP* (cit. on p. 103).

Rizos, Georgios, Konstantin Hemker, and Björn W. Schuller (2019). "Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification." In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. Ed. by Wenwu Zhu et al. ACM, pp. 991–1000. DOI: 10.1145/3357384.3358040. URL: https://doi.org/10.1145/3357384.3358040 (cit. on p. 101).

Schulz, Sarah et al. (2016). "Multimodular Text Normalization of Dutch User-Generated Content." In: *ACM TIST* 7.4, 61:1–61:22. DOI: 10.1145/2850422. URL: https://doi.org/10.1145/2850422 (cit. on p. 102).

Shridhar, Kumar et al. (2019). "Subword Semantic Hashing for Intent Classification on Small Datasets." In: *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*. IEEE, pp. 1–6. DOI: 10.1109/IJCNN.2019.8852420. URL: https://doi.org/10.1109/IJCNN.2019.8852420 (cit. on p. 100).

Socher, Richard et al. (2011). "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions." In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pp. 151–161. URL: https://www.aclweb.org/anthology/D11-1014/ (cit. on p. 111).

Sun, Xiao, Jiajin He, and Changqin Quan (2017). "A multi-granularity data augmentation based fusion neural network model for short text sentiment analysis." In: *Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos, ACII Workshops 2017, San*

Antonio, TX, USA, October 23-26, 2017. IEEE Computer Society, pp. 12–17. DOI: 10.1109/ACIIW.2017.8272616. URL: https://doi.org/10.1109/ACIIW.2017.8272616 (cit. on p. 102).

Toutanova, Kristina et al. (2001). "Text Classification in a Hierarchical Mixture Model for Small Training Sets." In: *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*. ACM, pp. 105–112. DOI: 10.1145/502585.502604. URL: https://doi.org/10.1145/502585.502604 (cit. on p. 100).

Vaswani, Ashish et al. (2017). "Attention is All you Need." In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf (cit. on p. 104).

Veliz, Claudia Matos, Orphée De Clercq, and Véronique Hoste (2019). "Benefits of Data Augmentation for NMT-based Text Normalization of User-Generated Content." In: *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019, Hong Kong, China, November 4, 2019*. Ed. by Wei Xu et al. Association for Computational Linguistics, pp. 275–285. DOI: 10.18653/v1/D19-5536. URL: https://doi.org/10.18653/v1/D19-5536 (cit. on p. 102).

Wang, Sida I. and Christopher D. Manning (2012). "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification." In: *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*. The Association for Computer Linguistics, pp. 90–94. URL: https://www.aclweb.org/anthology/P12-2018/ (cit. on p. 111).

Wei, Jason W. and Kai Zou (2019). "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui et al. Association for Computational Linguistics, pp. 6381–6387. DOI: 10.18653/v1/D19-1670. URL: https://doi.org/10.18653/v1/D19-1670 (cit. on p. 101).

Wu, Xing et al. (2019). "Conditional BERT contextual augmentation." In: *International Conference on Computational Science*. Springer, pp. 84–95 (cit. on p. 103).

Yu, Shujuan et al. (2019). "Hierarchical Data Augmentation and the Application in Text Classification." In: *IEEE Access* 7, pp. 185476–185485. DOI: 10.1109/ACCESS.2019.2960263. URL: https://doi.org/10.1109/ACCESS.2019.2960263 (cit. on p. 101).

6 Towards Authorship Attribution for Bibliometrics Using Stylometric Features

Andi Rexha,
Stefan Klampfl,
Mark Kröll,
Roman Kern

Abstract

The overwhelming majority of scientific publications are authored by multiple persons; yet, bibliographic metrics are only assigned to individual articles as single entities. In this paper, we aim at a more fine-grained analysis of scientific authorship. We therefore adapt a text segmentation algorithm to identify potential author changes within the main text of a scientific article, which we obtain by using existing PDF extraction techniques. To capture stylistic changes in the text, we adopt a number of stylometric features. We evaluate our approach on a small subset of PubMed articles consisting of an approximately equal number of research articles written by a varying number of authors. Our results indicate that the more authors an article has the more potential author changes are identified. These results can be considered as an initial step towards a more detailed analysis of scientific authorship, thereby extending the repertoire of bibliometrics.

6.1 Introduction

Bibliometrics has had to face the ever growing amount of scientific output in recent years – a challenge as well as a great opportunity. Techniques from other fields such as computer linguistics have been taken over (i) to speed up measuring processes as well as to (ii) to introduce novel ideas. In this paper we propose authorship attribution as additional method for bibliometrics. So far, authorship of a scientific article has been attributed to the given authors in a more or less unchallenged way. The extent of authorship is in general defined by community standards, for instance, it is in many scientific domains assumed that the lead author did most of the (writing) work and the last author contributed ideas being the head of the group. Applying authorship attribution methods enables us to attribute particular segments of an article to individual authors thereby analysing scientific authorship on a more fine- grained level. We would like to get more insights into writing style habits of scientists, for instance: Is there a preferred partitioning amongst authors? Is there a relation to the author ordering? In addition, these methods may also have the potential to measure whether the distribution of credit within a community or a research group is just.

As a first step into this direction, we seek to identify author changes within text passages. We thus apply TextSeqFault (Kern et al., 2012), an algorithm for intrinsic plagiarism detection - a line of research exhibiting a closely related problem setting. The algorithm was originally developed to detect changes in topics in order to apply text segmentation. To be applicable for authorship attribution, we adapted the algorithm to catch writing style changes by taking into account stylometric features. To evaluate our approach, we created a small subset of PubMed research articles. This data set consists of an approximately equal number of research articles for certain number of authors, ranging from one to four. In our experiments we could show that there exist a correlation between the number of authors and the stylomentric differences within the text.

6.2 Background

Coined by Pritchard et al., 1969, bibliometrics in general seeks to measure science by providing methods to explore, for example, the impact of a particular publication. Citation analysis (Garfield, 1972) represents one common method being an expression for simply counting a scientific article's citations which can be regarded as indicator for an article's scientific impact.

To face the ever growing amount of written publications, there was an increased interest in automating these methods by including ideas and techniques from other domains such as computer linguistics and network analysis. To that end, linguistic resources such as the ACL Anthology Reference Corpus (Bird et al., 2008) were compiled for standardization as well as comparison purposes with respect to research problems including reference analysis (Peng and McCallum, 2006), citation classification (Teufel, Siddharthan, and Tidhar, 2006) and generation of summaries (Elkiss et al., 2008). In this paper we introduce authorship attribution as an additional method for bibliometrics.

Authorship attribution (Stamatatos, 2009, Juola et al., 2008) expresses a classification setting where from a set of candidate authors, the author of a questioned article is to be selected. This line of research can be traced back to the 19th century, when Mendenhall, 1887 aimed to characterize the plays of Shakespeare. A century later (Mosteller and Wallace, 1964) used a Bayesian approach to analyse 'The Federalist Papers'. Since then, a line of research known as 'stylometry' focused on defining features to quantify an author's writing style Holmes, 1998 including (i) lexical features such as average word/sentence length and vocabulary richness, (ii) syntactical features such as frequency of function words and use of punctuation and (iii) structural features such as indentation. (Bergsma, Post, and Yarowsky, 2012) used stylometric features to detect the gender, native speaker vs. non-native speaker and conference vs. workshop paper.

6.3 Experimental Setup

Dataset For the evaluation we use a dataset composed of randomly selected documents from PubMed (http://www.ncbi.nlm.nih.gov/pubmed/), a free database created by the US National Library of Medicine holding full-text articles from the biomedical domain together with a standard XML mark-up that rigorously annotates the complete content of the published document, in particular the author metadata. The documents contained in this database are very diverse. In this work we focus on research articles only, but there is also a wide range of different article types, including book reviews and meeting reports.

For this evaluation we selected a small subset of the PubMed dataset consisting of an approximately equal number of research articles written by a certain number of authors, ranging from one to four. For our preliminary evaluation, we chose 10 research articles for each number of authors the BMC Bioinformatics journal – in total 40 articles.

PDF Extraction A prerequisite for the analysis of the writing style of scientific articles is the reliable extraction of their textual content. The portable document format (PDF), the most common format for scientific literature today, is optimised for presentation, but lacks structural information. As the raw character stream of the PDF is usually interrupted in mid-sentence by decorations or floating objects, extracting the main text of a scholarly article in the correct order requires the analysis of its document structure. To solve this task we build here upon our previous work (Klampfl et al., 2014), where we have developed an unsupervised processing pipeline that analyses the structure a PDF document using a number of both supervised and unsupervised machine learning techniques and heuristics. It processes a given PDF file in a sequence of individual processing modules and outputs the extracted body text. The first step builds upon the output of the Apache PDFBox library (http://pdfbox.apache.org) and uses unsupervised learning (clustering) to extract blocks of contiguous text from the raw PDF file and their column-wise reading order on each page. We consider these text blocks as the basic building blocks of a scientific article. In the next stage, these text blocks are categorized into different logical labels based on their

role within the document: meta-data blocks, decorations, figure and table captions, main text, and section headings. This stage is implemented as a sequential pipeline of detectors each of which labels a specific type of block. Apart from the meta-data detectors they are completely model-free and unsupervised. For more details on each of these detectors the interested reader is referred to (Klampfl et al., 2014). In the final stage of our PDF extraction pipeline the main body text of a scientific article is extracted by concatenating blocks containing section headings and main text in the reading order. Furthermore we resolve hyphenations at the end of lines and across blocks, columns, and pages.

Text Segmentation Our intrinsic plagiarism detection algorithm is based on a sliding window approach, originally developed for text segmentation. Text segmentation is applied in order to reconstruct individual document borders of a single, long document that was constructed by concatenating multiple textual documents, e.g., transcripts of spoken text. The majority of techniques for text segmentation are designed to detect changes in topics (Choi, 2000; Dias and Alves, 2005). Our text segmentation algorithm (Kern et al., 2012), named TextSeqFault, is a derivative of the well-known TextTiling algorithm, proposed by Hearst, 1997, and also falls into this category.

For each position within the document, preceding and succeeding consecutive sentences are combined into two adjacent sliding windows, which are then compared in a vector space. A dissimilarity measure calculates the relative difference between their inner similarity (the average pairwise similarity of sentences within the two windows) and their outer similarity (the average pairwise similarity of sentences across the two windows). This dissimilarity value is positive if the outer similarity is lower than the inner similarity, which indicates a potential topic change. The maximum value of 1 is reached if the outer similarity is zero, which is the case if the blocks correspond to orthogonal vectors. A topic change is reported when the dissimilarity exceeds a predefined threshold. As a similarity measure between two sentences we chose the common cosine similarity because of its simplicity and efficiency.

Stylometric Features In the original TextSeqFault algorithm (Kern et al., 2012) the features used to detect a change in topic are directly derived from the words within the sentences, i.e., by building a vector space of unigrams. We adapted the algorithm for the domain of intrinsic plagiarism detection by using a different set of features. Instead of topical features, such as word unigrams or other elements carrying semantic information, we made use of stylometric features, as we expected that topical features will be limited to work in cases where not only the authorship, but also the whole topic of the text dramatically changes. These stylometric features were chosen to reflect the style of the author, rather than the topic, which typically does not change within a single scientific article. In literature a wide array of stylometric features have been proposed (Mosteller and Wallace, 1964; Tweedie and Baayen, 1998; Stamatatos, 2009). Stylometric features have also been put to use in a number of use cases, e.g. for author profiling (Koppel, Argamon, and Shimoni, 2002) and vandalism detection (Harpalani et al., 2011). Table 7.1 shows the stylometric features used in our algorithm.

6.4 Evaluation

In order to produce a preliminary evaluation we decided to have a visual landscape of the dissimilarity within documents. For each of the analysed documents we calculate a stylometric dissimilarity among two adjacent sliding windows containing thirty sentences each. To show the results of this step in a larger scale, we multiply them with a scaling factor of up to 10.000. Furthermore we have normalized the length of the documents, where each position in the chart represent the dissimilarity of the relative position in the document.

Below we show two types of charts that aim to illustrate the style change among papers within the same category(with same number of authors) as well as a comparison among articles with different numbers of authors which aims to show a correlation between the number of authors and the dissimilarity of the writing style.

As illustrated in the Figure 6.1, there is a tendency of higher changes of writing style with the growing number of authors. The number of high

feature name	Description
alpha-chars-ratio	the fraction of total characters in the paragraph which are letters
digit-chars-ratio	the fraction of total characters in the paragraph which are digits
upper-chars-ratio	the fraction of total characters in the paragraph which are upper-case
white-chars-ratio	the fraction of total characters in the paragraph which are whitespace characters
type-token-ratio	ratio between the size of the vocabulary (i.e., the number of different words) and the total number of words
hapax-legomena	the number of words occurring once
hapax-dislegomena	the number of words occurring twice
yules-k	a vocabulary richness measure defined by Yule
simpsons-d	a vocabulary richness measure defined by Simpson
brunets-w	a vocabulary richness measure defined by Brunet
sichels-s	a vocabulary richness measure defined by Sichel
honores-h	a vocabulary richness measure defined by Honore
average-word-length	average length of words in characters
average-sentence-char-length	average length of sentences in characters
average-sentence-word-length	average length of sentences in words

Table 6.1: List of stylometric features used in our text segmentation algorithm. Many of those features are defined in (Tweedie and Baayen, 1998).

peaks (which represent a big change of the writing style) grows with the growing of the amount of the authors for the paper.

The inspection of the Figure 6.2 highlights the differences between papers written by different amount of authors. The papers with one and two authors tend to have a flat shape showing a small dissimilarity within the document. On the other hand the papers with three and four authors are inclined to have bigger and larger variations of writing style. In a closer look, also the document with four authors shows the tendency of higher number of large dissimilarity compared to the three authors paper.

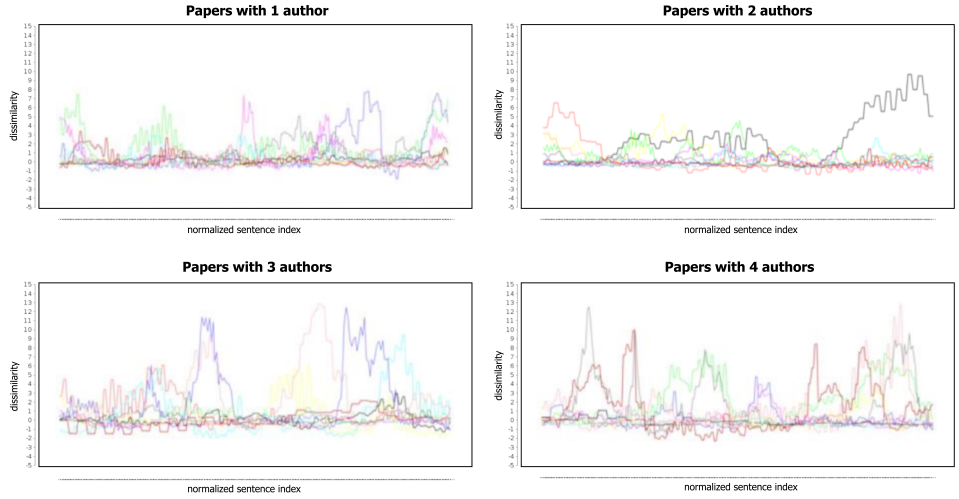


Figure 6.1: Landscape of the writing style dissimilarity for papers with different number of authors.

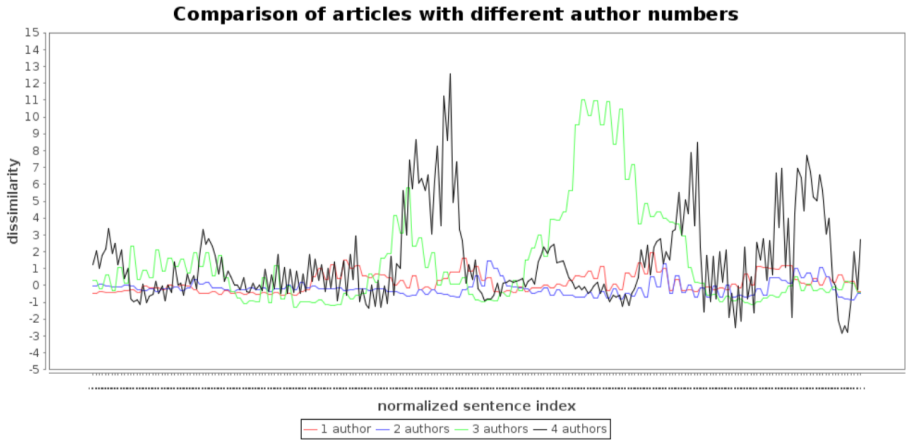


Figure 6.2: Comparison of writing style dissimilarity among papers with different number of authors.

6.5 Conclusion

In this paper, we proposed to add authorship attribution methods to the repertoire of bibliometrics thereby enabling a more fine-grained analysis of authorship. As a first step into this direction we presented an algorithm to

segment scientific articles according to writing style changes. Our preliminary results corroborate the natural assumption that in most cases the more authors contribute the more author changes are identified. In future work, we will extend our evaluation to more articles across topics as well as across journals. In addition, we intend to learn classification models for individual authors capturing the respective writing style trying to associate each part to the individual author. This feature might be used to credit differently the contribution of each author to the paper.

Bibliography

Bergsma, Shane, Matt Post, and David Yarowsky (2012). "Stylometric analysis of scientific articles." In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 327–337 (cit. on p. 127).

Bird, Steven et al. (2008). "The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics." In: (cit. on p. 127).

Choi, Freddy YY (2000). "Advances in domain independent linear text segmentation." In: *arXiv preprint cs/0003083* (cit. on p. 129).

Dias, Gael and Elsa Alves (2005). "Unsupervised topic segmentation based on word co-occurrence and multi-word units for text summarization." In: *Proceedings of the ELECTRA Workshop associated to 28th ACM SIGIR Conference, Salvador, Brazil*, pp. 41–48 (cit. on p. 129).

Elkiss, Aaron et al. (2008). "Blind men and elephants: What do citation summaries tell us about a research article?" In: *Journal of the American Society for Information Science and Technology* 59.1, pp. 51–62 (cit. on p. 127).

Garfield, Eugene (1972). "Citation analysis as a tool in journal evaluation." In: *Science* 178.4060, pp. 471–479 (cit. on p. 127).

Harpalani, Manoj et al. (2011). "Language of vandalism: Improving wikipedia vandalism detection via stylometric analysis." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pp. 83–88 (cit. on p. 130).

Hearst, Marti A (1997). "TextTiling: Segmenting text into multi-paragraph subtopic passages." In: *Computational linguistics* 23.1, pp. 33–64 (cit. on p. 129).

Holmes, David I (1998). "The evolution of stylometry in humanities scholarship." In: *Literary and linguistic computing* 13.3, pp. 111–117 (cit. on p. 127).

Juola, Patrick et al. (2008). "Authorship attribution." In: *Foundations and Trends® in Information Retrieval* 1.3, pp. 233–334 (cit. on p. 127).

Kern, Roman et al. (2012). "Teambeam-meta-data extraction from scientific literature." In: *D-Lib Magazine* 18.7, p. 1 (cit. on pp. 126, 129, 130).

Klampfl, Stefan et al. (2014). "Unsupervised document structure analysis of digital scientific articles." In: *International journal on digital libraries* 14.3-4, pp. 83–99 (cit. on pp. 128, 129).

Koppel, Moshe, Shlomo Argamon, and Anat Rachel Shimoni (2002). "Automatically categorizing written texts by author gender." In: *Literary and linguistic computing* 17.4, pp. 401–412 (cit. on p. 130).

Mendenhall, Thomas Corwin (1887). "The characteristic curves of composition." In: *Science* 9.214, pp. 237–249 (cit. on p. 127).

Mosteller, Frederick and David Wallace (1964). *Inference and disputed authorship: The Federalist.(1964)* (cit. on pp. 127, 130).

Peng, Fuchun and Andrew McCallum (2006). "Information extraction from research papers using conditional random fields." In: *Information processing & management* 42.4, pp. 963–979 (cit. on p. 127).

Pritchard, Alan et al. (1969). "Statistical bibliography or bibliometrics." In: *Journal of documentation* 25.4, pp. 348–349 (cit. on p. 127).

Stamatatos, Efstathios (2009). "A survey of modern authorship attribution methods." In: *Journal of the American Society for information Science and Technology* 60.3, pp. 538–556 (cit. on pp. 127, 130).

Teufel, Simone, Advaith Siddharthan, and Dan Tidhar (2006). "Automatic classification of citation function." In: *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 103–110 (cit. on p. 127).

Tweedie, Fiona J and R Harald Baayen (1998). "How variable may a constant be? Measures of lexical richness in perspective." In: *Computers and the Humanities* 32.5, pp. 323–352 (cit. on pp. 130, 131).

7 Towards a More Fine Grained Analysis of Scientific Authorship: Predicting the Number of Authors Using Stylometric Features

Andi Rexha,
Stefan Klampfl,
Mark Kröll,
Roman Kern

Abstract

To bring bibliometrics and information retrieval closer together, we propose to add the concept of author attribution into the pre-processing of scientific publications. Presently, common bibliographic metrics often attribute the entire article to all the authors affecting author-specific retrieval processes. We envision a more fine- grained analysis of scientific authorship by attributing particular segments to authors. To realize this vision, we propose a new feature representation of scientific publications that captures the distribution of stylometric features. In a classification setting, we then seek to predict the number of authors of a scientific article. We evaluate our approach on a data set of $\tilde{6}$100 PubMed articles and achieve best results by applying random forests, i.e., 0.76 precision and 0.76 recall averaged over all classes.

7.1 Introduction

The ongoing growth of the volume of scholarly publications poses significant challenges to both information retrieval processes in digital libraries as well as bibliometric techniques that analyse academic literature in a quantitative manner. Ideas from other fields such as computer linguistics have been incorporated into bibliometrics to improve and enhance the measuring and analysis processes. To bring bibliometrics and information retrieval closer together, we propose to add the concept of author attribution into the pre-processing of the analysis of scientific publications. Yet, since common bibliographic metrics often attribute the entire article to all the authors, we introduce a reinterpretation of authorship attribution: to attribute particular segments of an article to individual authors allowing for a more fine-grained analysis of contribution and role. Information retrieval systems could then benefit from such authorship attribution in the following ways: Scholarly search engines could implement an author specific search which allows researchers to specifically look for text passages written by a particular author. This more precise passage-author attribution then allows the generation of researcher profiles. These profiles would reflect a researcher's contributions to different scientific fields in a more detailed manner. In addition, the profile might be valuable for predicting and thus understanding a researcher's role, for example, more actively involved (writing) vs. acting more like a mentor providing ideas and giving feedback (less involved in writing; reflected for example by author positioning).

As a first step in this direction we have recently applied text segmentation to identify potential author changes within the main text of a scientific article (Rexha et al., 2015). We have adopted a number of stylometric features to capture stylistic changes in the text, following the hypothesis that different authors manifest in different writing styles within the document. In this work we extend this work by applying a new feature representation of scientific documents that captures the distribution of stylometric features across the document and to predict the number of authors accordingly. The classification performance then represents so- to-say a quantification of the amount of information that is contained within the stylometry of a scientific article about the number of authors involved in writing it.

The text for the analysis is produced by a PDF processing pipeline, which analyses scientific articles and extracts, among other information, also the main text (Klampfl et al., 2014). As training data we have chosen a subset of PubMed research articles. This data set consists of a wide variety of journals across different domains. We have selected an approximately equal number of research articles written by a certain number of authors, ranging from one to five. This work is structured as follows: First, we elaborate on existing work on authorship attribution techniques as well as the retrieval of higher level knowledge from scientific texts in general. Then, we describe our experimental setup, including the dataset and extracted stylometric features. Finally, we present our results and give an outlook for future work.

7.2 Related Work

Over the past decades one can observe an ever growing amount of scientific output; much to the joy of research areas such as (i) Bibliometrics which applies statistics to measure scientific impact and (ii) Information Retrieval which applies natural language processing to make the valuable body of knowledge accessible. This interest in processing and exploiting scientific publications from different perspectives is reflected by venues such as the International Workshop on Bibliometric-enhanced Information Retrieval (Mayr et al., 2014), the International Workshop on Mining Scientific Publications[1] or Mining Scientific Papers: Computational Linguistics and Bibliometrics[2].

To be of value for both fields, scientific publications need to be semantically enriched. Adding semantics includes assigning instances to concepts which are organized and structured in dedicated ontologies. Entity and relation recognition thus represent a vital pre-processing step. To give an example, medical entity recognition (Abacha and Zweigenbaum, 2011) seeks to extract

[1]Conference: Proceedings of the 4th Workshop on Mining Scientific Publications. Co-located with the Joint Conference on Digital Libraries (JCDL), Knoxville, Tennessee, 2015.

[2]Conference: Proceedings of the First Workshop on Mining Scientific Papers: Computational Linguistics and Bibliometrics co-located with 15th International Society of Scientometrics and Informetrics Conference (ISSI), Istanbul, Turkey, 2015.

instances from classes such as "Disease", "Symptom" or "Drug" to enrich the retrieval process. Research assistants such as BioRAT (Corney et al., 2004) or FACTA (Tsuruoka, Tsujii, and Ananiadou, 2008) then can offer an added value employing this type of semantic information.

Departing from a mere content-level, Liakata et al., 2012 introduced a different approach by focusing on the discourse structure to characterize the knowledge conveyed within the text. For this purpose, the authors identified 11 core scientific concepts including "Motivation", "Result" or "Conclusion". Ravenscroft, Liakata, and Clare, 2013 present the Partridge system which automatically categorizes articles according to their types such as "Review" or "Case Study". In a similar manner, the TeamBeam (Kern et al., 2012) algorithm extracts structured meta-data, such as the title, journal name and abstract, as well as information about the article's authors.

In this work we introduce the concept of authorship attribution as an additional pre- processing step for subsequent retrieval procedures. Authorship attribution, in general, expresses a classification setting where from a set of candidate authors the author of a questioned article is to be selected (Stamatatos, 2009; Juola et al., 2008). This line of research can be traced back to the 19th century, when Mendenhall, 1887 aimed to characterize the plays of Shakespeare. A century later Mosteller and Wallace, 1964 used a Bayesian approach to analyse 'The Federalist Papers'. Since then, a line of research known as stylometry focused on defining features to quantify an author's writing style (Holmes, 1998). Bergsma, Post, and Yarowsky, 2012 used stylometric features to detect the gender of an author and to distinguish between native vs. non-native speakers and conference vs. workshop papers. In this work, we use stylometric features to classify scientific papers according to the number of its authors.

7.3 Experimental Setup

Dataset For the evaluation we use a dataset composed of randomly selected documents from PubMed (`http://www.ncbi.nlm.nih.gov/pubmed/`), a free database created by the US National Library of Medicine holding full-text articles from the biomedical domain together with a standard XML

mark-up that rigorously annotates the complete content of the published document, in particular the author metadata. The documents contained in this database are very diverse. In this work we limit ourselves to research articles only, but there is also a wide range of different article types, including book reviews and meeting reports. For this evaluation we selected a subset of the PubMed dataset consisting of an approximately equal number of research articles written by a certain number of authors, ranging from one to five. For our evaluation, we chose 6144 research articles in total, across 563 different journals and publication entities. There were 983, 1192, 1391, 1418, and 1160 articles with one, two, three, four, and five authors, respectively.

PDF Extraction A prerequisite for the writing style analysis of scientific articles is the reliable extraction of their textual content. The portable document format (PDF), the most common format for scientific literature today, is optimised for presentation, but lacks structural information. As the raw character stream of the PDF is usually interrupted in mid-sentence by decorations or floating objects, extracting the main text of a scholarly article in the correct order requires the analysis of its document structure. To solve this task we build here upon our previous work (Kern et al., 2012; Klampfl et al., 2014), where we have developed a processing pipeline that analyses the structure a PDF document using a number of both supervised and unsupervised machine learning techniques and heuristics. It processes a given PDF file in a sequence of individual processing modules and outputs the extracted body text. The first step builds upon the output of the Apache PDFBox library (http://pdfbox.apache.org) and uses unsupervised learning (clustering) to extract blocks of contiguous text from the raw PDF file and their column-wise reading order on each page. We consider these text blocks as the basic building blocks of a scientific article. In the next stage, these text blocks are categorized into different logical labels based on their role within the document: meta-data blocks, decorations, figure and table captions, main text, and section headings. This stage is implemented as a sequential pipeline of detectors each of which labels a specific type of block. Apart from the meta-data detectors they are completely model-free and unsupervised. For more details on each of these detectors the interested reader is referred to (Klampfl et al., 2014). In the final stage of our PDF extraction pipeline the main body text of a scientific article is extracted

by concatenating blocks containing section headings and main text in the reading order. We resolve hyphenations at the end of lines and across blocks, columns, and pages. Furthermore, paragraphs that span more than one column or page are merged.

Stylometric features and document representation Capturing different writing styles within a document requires the extraction and analysis of suitable features. Topical features, such as word unigrams or other elements carrying semantic information, are helpful in identifying document segments which differ not only in the author, but also in the whole topic of the text. On the other hand, stylometric features reflect the author's writing style, rather than the topic, which typically does not change within a single scientific article, and generalizes across different domains. To compare and classify different scientific articles based on the number of authors involved, we try to capture the distribution of stylometric features across a single document. We split he document into continuous segments (here a segment corresponds to a sentence) and extract the stylometric features for each of those segments. We then view the document as a distribution of different stylometric features. The literature suggests a broad amount of stylometric features (Mosteller and Wallace, 1964; Tweedie and Baayen, 1998; Stamatatos, 2009). Table 7.1 presents the list of features we extract for each segment. In addition, we calculate the minimum, maximum, average and variance for each of those features across every document.

7.4 Evaluation

In order to evaluate whether the stylometric feature representation of scientific articles contains authorship information, we trained different classifiers in a supervised manner to predict the number of authors for each document. From the articles in the PubMed dataset, we extracted the stylometric features for each sentence of the document and represented the distribution of this features across the document as its maximum, minimum, average and variance. As a further preprocessing step we normalized these feature values to avoid dominating features in the learning process. For our experiments,

feature name	Description
alpha-chars-ratio	the fraction of total characters in the paragraph which are letters
digit-chars-ratio	the fraction of total characters in the paragraph which are digits
upper-chars-ratio	the fraction of total characters in the paragraph which are upper-case
white-chars-ratio	the fraction of total characters in the paragraph which are whitespace characters
type-token-ratio	ratio between the size of the vocabulary (i.e., the number of different words) and the total number of words
hapax-legomena	the number of words occurring once
hapax-dislegomena	the number of words occurring twice
yules-k	a vocabulary richness measure defined by Yule
simpsons-d	a vocabulary richness measure defined by Simpson
brunets-w	a vocabulary richness measure defined by Brunet
sichels-s	a vocabulary richness measure defined by Sichel
honores-h	a vocabulary richness measure defined by Honore
average-word-length	average length of words in characters
average-sentence-char-length	average length of sentences in characters
average-sentence-word-length	average length of sentences in words

Table 7.1: List of stylometric features used in our text segmentation algorithm. Many of those features are defined in (Tweedie and Baayen, 1998).

we selected two classification algorithms: Logistic regression and Random Forest. Table 7.2 and Table 7.3 report individual class results achieved by Logistic Regression and Random Forest algorithms. Comparing the two classification algorithms, we notice that the Random Forest outperforms the Logistic Regression algorithm by far. As can be seen, both algorithms achieve the lowest performance in predicting the 5-authors class. We believe that this outcome might be due to two different aspects. The first aspect has to do with the amount of contribution from each author. The smaller the amount of text an author writes, the more difficult to distinguish it from the contribution of other authors. The second aspect relates to the actual writing contributions. We think that the larger the amount of writers the more likely is that some of them may not have contributed at all in the writing of the paper.

Another consideration that we can make relates to the 1-author class. The

Class/Metric	Precision	Recall	F-Measure
Class 1-author	0.533	0.482	0.506
Class 2-authors	0.330	0.301	0.315
Class 3-authors	0.369	0.522	0.432
Class 4-authors	0.235	0.432	0.394
Class 5-authors	0.235	0.105	0.145
Average	0.365	0.376	0.362

Table 7.2: Performance of classifying the number of authors of a scientific article using logistic regression on our dataset (10-fold cross-validation).

Class/Metric	Precision	Recall	F-Measure
Class 1-author	0.881	0.780	0.827
Class 2-authors	0.755	0.681	0.716
Class 3-authors	0.759	0.801	0.780
Class 4-authors	0.724	0.796	0.759
Class 5-authors	0.687	0.699	0.693
Average	0.759	0.755	0.755

Table 7.3: Performance of classifying the number of authors of a scientific article using random forests on our dataset (10-fold cross-validation).

performance of both algorithms exceeds the results for the other classes. We believe that this is due to the correctness of the data. The 1-author papers are less likely to have more authors than mentioned in the paper, making the data more representative for this class.

7.5 Conclusion

In this paper, we classified scientific articles according to their number of authors by using a set of stylometric features. We applied supervised learning to this setup and achieved best results with Random Forests. The classification results suggest that the stylometric feature space in fact captures variations in the writing style that we would expect from multiple contributing authors.

This work fosters our understanding towards a more fine-grained analysis

of scientific authorship by attributing particular segments to authors. Information retrieval systems could benefit from this concept of authorship attribution, for instance, in course of author specific search.

Bibliography

Abacha, Asma Ben and Pierre Zweigenbaum (2011). "Medical entity recognition: A comparison of semantic and statistical methods." In: *Proceedings of BioNLP 2011 Workshop*. Association for Computational Linguistics, pp. 56–64 (cit. on p. 139).

Bergsma, Shane, Matt Post, and David Yarowsky (2012). "Stylometric analysis of scientific articles." In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 327–337 (cit. on p. 140).

Corney, David PA et al. (2004). "BioRAT: extracting biological information from full-length papers." In: *Bioinformatics* 20.17, pp. 3206–3213 (cit. on p. 140).

Holmes, David I (1998). "The evolution of stylometry in humanities scholarship." In: *Literary and linguistic computing* 13.3, pp. 111–117 (cit. on p. 140).

Juola, Patrick et al. (2008). "Authorship attribution." In: *Foundations and Trends® in Information Retrieval* 1.3, pp. 233–334 (cit. on p. 140).

Kern, Roman et al. (2012). "Teambeam-meta-data extraction from scientific literature." In: *D-Lib Magazine* 18.7, p. 1 (cit. on pp. 140, 141).

Klampfl, Stefan et al. (2014). "Unsupervised document structure analysis of digital scientific articles." In: *International journal on digital libraries* 14.3-4, pp. 83–99 (cit. on pp. 139, 141).

Liakata, Maria et al. (2012). "Automatic recognition of conceptualization zones in scientific articles and two life science applications." In: *Bioinformatics* 28.7, pp. 991–1000 (cit. on p. 140).

Mayr, Philipp et al. (2014). "Bibliometric-enhanced information retrieval." In: *European Conference on Information Retrieval*. Springer, pp. 798–801 (cit. on p. 139).

Mendenhall, Thomas Corwin (1887). "The characteristic curves of composition." In: *Science* 9.214, pp. 237–249 (cit. on p. 140).

Mosteller, Frederick and David Wallace (1964). *Inference and disputed authorship: The Federalist.(1964)* (cit. on pp. 140, 142).

Ravenscroft, James, Maria Liakata, and Amanda Clare (2013). "Partridge: An effective system for the automatic cassification of the types of academic papers." In: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, pp. 351–358 (cit. on p. 140).

Rexha, Andi et al. (2015). "Towards Authorship Attribution for Bibliometrics using Stylometric Features." In: *CLBib@ ISSI*, pp. 44–49 (cit. on p. 138).

Stamatatos, Efstathios (2009). "A survey of modern authorship attribution methods." In: *Journal of the American Society for information Science and Technology* 60.3, pp. 538–556 (cit. on pp. 140, 142).

Tsuruoka, Yoshimasa, Juníchi Tsujii, and Sophia Ananiadou (2008). "FACTA: a text search engine for finding associated biomedical concepts." In: *Bioinformatics* 24.21, pp. 2559–2560 (cit. on p. 140).

Tweedie, Fiona J and R Harald Baayen (1998). "How variable may a constant be? Measures of lexical richness in perspective." In: *Computers and the Humanities* 32.5, pp. 323–352 (cit. on pp. 142, 143).

8 Authorship Identification of Documents with High Content Similarity

Andi Rexha,
Mark Kröll,
Hermann Ziak,
Roman Kern

Abstract

The goal of our work is inspired by the task of associating segments of text to their real authors. In this work, we focus on analyzing the way humans judge different writing styles. This analysis can help to better understand this process and to thus simulate/ mimic such behavior accordingly. Unlike the majority of the work done in this field (i.e. authorship attribution, plagiarism detection, etc.) which uses content features, we focus only on the stylometric, i.e. content-agnostic, characteristics of authors. Therefore, we conducted two pilot studies to determine, if humans can identify authorship among documents with high content similarity. The first was a quantitative experiment involving crowd-sourcing, while the second was a qualitative one executed by the authors of this work. Both studies confirmed that this task is quite challenging. To gain a better understanding of how humans tackle such a problem, we conducted an exploratory data analysis on the results of the studies. In the first experiment, we compared the decisions against content features and stylometric features. While in the second, the

evaluators described the process and the features on which their judgment was based. The findings of our detailed analysis could (1) help to improve algorithms such as automatic authorship attribution as well as plagiarism detection, (2) assist forensic experts or linguists to create profiles of writers, (3) support intelligence applications to analyze aggressive and threatening mes- sages and (4) help editor conformity by adhering to, for instance, journal specific writing style.

8.1 Introduction

Identifying and attributing authorship of different text passages can be beneficial for various tasks and areas including bibliometrics, information retrieval and plagiarism detection. In previous work, we illustrated how to incorporate ideas of authorship attribution into the mentioned areas. We introduced the concept of extending the retrieval process by including authorship information to allow for identifying text passages written by a particular author. In Rexha et al., 2015, we applied text segmentation to identify potential author changes within the main text of a scientific article. Rexha et al., 2016 presented a new feature representation of scientific documents that capture the distribution of stylometric features across the document and to predict the number of authors accordingly.

This follow-up work (an extension of Rexha et al., 2016) investigates the extent with which content-agnostic, stylometric features are capable of distinguishing between authors. We analyze the way human evaluators[1] judge and assign similar writing styles with respect to text passages exhibiting a high content similarity. In contrast to related work, we thus focus on pure stylistic characteristics of authors, which usually tend to generalize better if the topic changes. Consider following two examples:

Example 8.1. *Stylometry is the application of the study of linguistic style, usually to written language.*

Example 8.2. *Stylometry is the application of the study of linguistic style. It is usually applied to written language.*

[1] In this work, we use "user", "evaluator" and "annotator" as synonyms

The information of these two sentences is the same, yet they slightly differ in the way, how this information is conveyed. In Example 8.1, the author appears to favor longer sentences; in Example 8.2 shorter sentences. In case of these constructed examples content features will not be sufficient to distinguish between authors; to do that we need to focus on content-agnostic, i.e. pure stylometric features. Consider the case of a journal hiring a new employee and asking her to adhere to a (journal-) specific style in her articles. Providing a tool that compares her writing style to the journal style can shorten the settling in process. The same logic applies to intelligence agencies to support the analysis of aggressive and threatening messages, i.e. identify the responsible author. Furthermore, commercial products (i.e. Grammarly[2]) that help in writing can incorporate such mechanisms and thus assist authors to keep the same writing style throughout the document. Finally, these findings can also help to build systems supporting forensic experts with suggestions for candidates of plagiarism.

As a first step towards building such a system, we conducted two different pilot studies to understand whether humans are capable of distinguishing between writing styles without considering the content as a discriminating feature. As already mentioned, this makes our study different from most of plagiarism detection tasks. The first is a quantitative study with annotators from the crowdsourcing platform CrowdFlower. This study has the disadvantage of the unknown quality of the judgments. Even though we use mechanisms to avoid random choices from the crowd, we cannot fully prevent it. To overcome such a problem, we performed a qualitative study, which represents the main contribution of this article. In this study, the current authors evaluate the list of experiments and give a detailed description of the thinking process while annotating.

The outcome of the two studies was quite similar. Even though it appears more likely not to have random evaluations (72% confidence), our findings revealed that this task is challenging, even for humans. We also conducted an exploratory data analysis where we statistically compared the decisions against content features and content-agnostic features. We didn't find any correlation that explains the different judgments between annotators. In

[2] https://www.grammarly.com/

addition, we make our dataset publicly available[3].

8.2 Related Work

Over the past decades, we have observed an ever-growing amount of scientific output; much to the joy of research areas such as bibliometrics as well as scientometrics which both aim to measure and quantify the scientific output. This ongoing growth of the volume of scholarly publications poses significant challenges leading to the incorporation of ideas from other fields such as computer linguistics to improve and enhance the measuring and analysis processes. In recent work, we proposed to add the concept of author attribution into the pre-processing of the analysis of scientific publications. In Rexha et al., 2015, we focused on attributing particular segments of an article to individual authors. We thereby initiated a discussion on the implicit definition of scientific authorship; to give an example, in many scientific domains it is assumed that the first author did most of the (writing) work and the last author contributed ideas being the head of the research group. Follow-up work (Rexha et al., 2016) studied the distribution of stylometric features across a scientific article to predict the number of authors accordingly. The classification performance then represents so-to-say a quantification of the amount of information that is contained within the stylometry of a scientific article about the number of authors involved in writing it.

Authorship attribution (Stamatatos, 2009; Juola et al., 2008) can be expressed as classification task where, from a set of candidate authors the author of a disputed article is to be identified. This line of research can be traced back to the 19th century when Mendenhall, 1887 aimed to characterize the plays of Shakespeare. A century later Mosteller and Wallace, 1964 used a Bayesian approach to analyze 'The Federalist Papers'—one of the first authorship disputes in literature. Since then, a line of research known as stylometry focused on defining features to quantify an author's writing style (Holmes, 1998) including (1) lexical features such as average word/sentence length

[3] https://zenodo.org/record/437461

and vocabulary richness, (2) syntactical features such as frequency of function words and use of punctuation, and (3) structural features such as indentation. In terms of supervised classification, it translates into the task of proper feature selection/extraction (Stamatatos, 2009). This process of selecting features is often closely related to the research or application scenario at hand, i.e. adapted to domains, genres or textual characteristics. Attributing authors being faced with short messages (Villar-Rodriguez et al., 2016; Brocardo et al., 2013) differs from being faced with unstructured texts (Zhang et al., 2014).

With the advent of social media, in particular, the way our society communicates and exchanges information has changed. Social media opens up new opportunities to express opinion. The process of determing authors of online messages, especially those with offensive as well as threatening expressions, is thus given higher priority. Silva et al., 2011 propose a set of stylistic markers for automatically attributing authorship to micro blogging messages such as Twitter. In their classification setting, they are investigating whether 'non-traditional', content-agnostic markers such as emoticons contain relevant information for the task. Inches, Harvey, and Crestani, 2013 and Knaap and Grootjen, 2007 conduct authorship attribution analyses in chat logs exploring statistical approaches as well as formal concept analysis respectively. With respect to cybercrime, Iqbal et al., 2013 experiment with different settings including the characterization of authorship. They present a data mining approach that uses frequent stylometric patterns, i.e. a combination of stylometric feature items that occurs frequently.

Adding information on authorship can also be considered as adding semantic information, thus supporting the analysis of scientific publications. To be of value, scientific publications are subjected to semantic enrichment in various ways. Adding semantics includes, for instance, assigning instances to concepts which are organized and structured in dedicated ontologies. Entity and relation recognition thus represent a vital pre-processing step. To give an example, medical entity recognition (Abacha and Zweigenbaum, 2011) seeks to extract instances from classes such as "Disease", "Symptom" or "Drug" to enrich the retrieval process. Research assistants such as BioRAT (Corney et al., 2004) or FACTA (Tsuruoka, Tsujii, and Ananiadou, 2008) then can offer an added value employing this type of semantic information. Liakata et al., 2012 departed from a mere content-level enrichment

and focused on the discourse structure to characterize the knowledge conveyed within the text. For this purpose, they identified 11 core scientific concepts including "Motivation", "Result" or "Conclusion". In the Partridge system, Ravenscroft, Liakata, and Clare, 2013 built upon the automated recognition to automatically categorize articles according to their types such as Review or Case Study. The TeamBeam (Kern et al., 2012) algorithm aims to extract an article's meta-data, such as the title, journal name and abstract, as well as explicit information about the article's authors. Implicit information about an author includes her writing style, which reflects among others, the writer's personality as well as directly relates to characteristics such as readability and clarity. Stylometry represents the line of research which focuses on defining features to quantify an author's writing style (Holmes, 1998). Bergsma, Post, and Yarowsky, 2012 used stylometric features to detect the gender of an author and to distinguish between native versus non-native speakers and conference versus workshop papers.

8.3 Experimental Setup

In order to understand whether humans can identify the authorship once the content information has been removed, we conducted two pilot studies. In these studies, we provided human annotators with one source and four target textual snippets in different experiments. In the first experiment, one of the targets is written by the same author as the source, and the other three are written by different authors as the source. Then, we have the annotators rank the snippets from the most to the least similar concerning the writing style, asking them to classify the target written by the same author as "most similar" (see Figure8.1). Since we wanted to extract all the clues about the stylometry, we forced users to rank the articles avoiding to provide them with options like "not able to find".

For the studies, we selected data from Pubmed[4], a free database created by the US National Library of Medicine. This database holds full-text articles from the biomedical domain together with a standard XML markup that rigorously annotates the complete content of the published document. It also

[4] https://www.ncbi.nlm.nih.gov/pubmed

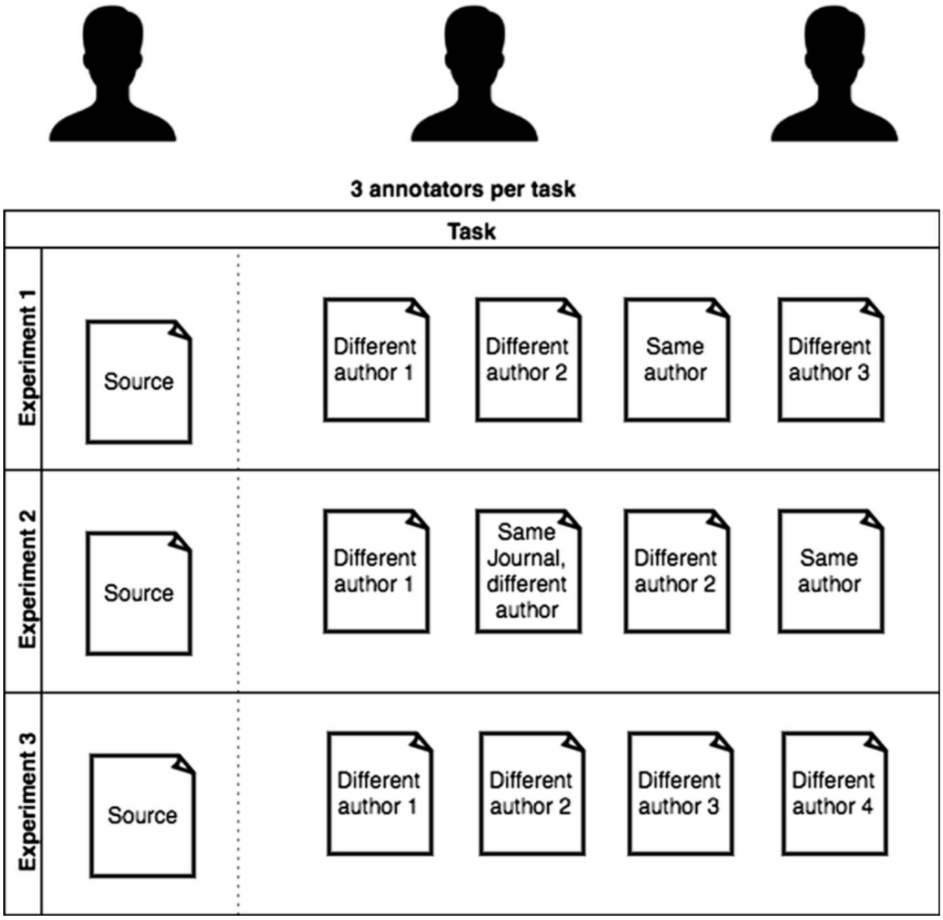


Figure 8.1: Description of a task. Three evaluators are assigned to each task and rank each target of the experiment from the "most similar" to the "least similar".

contains valuable metadata about the authors and the journal in which the article is published. At first, we retrieve documents written by only a single author to obtain pure writing styles. Note that some articles could be written by ghostwriters or by colleagues of authors helping them with English writing. In some other cases, the institution provides the authors with editing services, blurring the real style of the authors. Another drawback of this hypothesis relies on the possible change of the writing style in distance of years. We intend to address such shortcomings in our future work.

From the previously selected documents, we chosen a subset and decided to make the annotators rank text snippets which are drawn from the beginning of the introduction section (we select the first sentences until the one ending after the 400-th character). Therationale behind this choice is twofold: (1) it gets more and more difficult for the user to remain focused on the task while reading a long text; (2) we hypothesize that the introduction contains less topological information than other parts of the scientific papers. Having selected the text snippets, we designed three experiments (which we call a task) for each annotator. For each of the experiments we presented the annotators with a source and four target snippets, subject to different problem settings (see Figure 8.1)

- In Experiment 1, we presented a target snippet written by the same author as the source as well as three others written by different authors as the source.
- In Experiment 2, we provided the annotators with one target article written by the same author as the source, one target article from a different author but published in the same journal as the source and two target articles written by different authors and published in different journals as the source. This experiment is designed to capture any correlation between the writing style within the same journal, presumably within the same scientific topic.
- In Experiment 3, we wanted to gain as much information as possible from the users' thinking while ranking. Thus, we showed four target snippets written by different authors as the one of the source snippet, while still suggesting to the annotator that one of the targets is written by the same author as the source.

In the last design step of our pilot study, we selected 90 random snippets from the PubMed database as candidate source snippets. We indexed the database of the snippets from single authors by stemming the words and removing the stop-words. We assigned 30 snippets to each of the categories of the experiments, and we performed for each of them a search according to the request:

- In Experiment 1, we searched for 10 similar articles from the same author and 100 from different authors.

- In Experiment 2, we searched for 10 similar articles from the same author, 10 from the same journal but different author, and 100 from different authors and journals.
- In Experiment 3, we searched for 100 similar articles from different authors.

Based on these results, we performed a cosine similarity between the vector of the words with the source snippet and selected the most similar ones according to the experiment description. This way, the content information should have been removed as a source of information for authorship identification. For example, for Experiment 1 we selected the most similar article from the same author and three from different authors. Additionally, we also performed a manual check and removed the snippets that we assumed contained content hints for texts written by the same author (mainly based on keywords or phrases). At the end of this phase, we chose 66 experiments (22 per each category of experiments previously described).

Once we build the experiments, we made two different studies. The first was quantitative (due to the vast amount of users), performed using the crowd-sourcing platform CrowdFlower[5]. The platform provides workforce from different countries helping to label and to enrich data. Here, we presented the same set of experiments (task) to three different annotators (see Figure 8.1). Although we have used some mechanisms to avoid random selection, it is almost impossible to assure it.

In the second study, we perform a qualitative experiment, where the authors of this work evaluated all the experiments. We call it qualitative due to the quality assurance from the annotators. Three of the users, were asked to perform a stylometric ranking, and the forth to focus on the content. Furthermore, we split the set of the experiments in two equal parts. In the first part, we presented to the user the same experiments as shown to the crowd. In the second one, we added a list of features that might help the evaluators with the ranking. We extracted the following features for each of the text snippets:

- Numeric Features, which contain numeric information about the writing style. We used the features listed in the Table 8.2. We augmented

[5] https://www.crowdflower.com/

this list with the "depth of the dependency trees"(indicating the maximum depth of the dependency tree in each sentence) as well as with "stop words ratio" (indicating the ratio of the stop words with the total number of tokens in each sentence).
- Common selected phrases, representing a list of selected phrases present in both, source and target snippet, like "although", "most common", etc.
- Descriptive features, informing whether the first sentence of the source and the target starts with a definition (for example: "Breast cancer is...") or whether hyphens are used in both snippets.
- Outliers, representing a list of Numeric Features that have a larger value than the 95th percentile and smaller than the 5th percentile of the distribution in all the snippets of the experiments.

Then, we presented them to the users (see Figure 8.2). For the numeric features, for each target, we showed to the user the closest value with the source. We indicate them as "Winners". We also present the list of "Common selected" phrases and "Descriptive features". Lastly, we showed the Outliers denoting with "!" the features with smaller values than the 5th percentile and with "^" for values with larger values than the 95th percentile of the distribution.

8.4 Pilot Study #1

The job in CrowdFlower was performed by 56 different annotators from 29 different countries. Since our goal is to rank based on the writing style, the level of understanding English isn't of a big concern for the task. To avoid random selection, we configure the system to disallow annotation in less than 20 seconds. At first glance, the annotators have a small agreement in the ranking of the similarity between source and target snippets. Without considering the rank itself, a full agreement is achieved in 26 targets, 160 have an agreement of two annotators, and 78 of the targets have no agreement at all. For a more detailed analysis, we use Krippendorff's alpha (Krippendorff, 2004) measure to determine the inter-rater agreement for the ranking of

```
Target-4:
Low back pain (LBP) is one of the most common public health problems in modern
industrialized societies. Many lumbar problems are muscular in origin and
persons suffering from LBP often have weak lumbar muscles []. Many studies have
suggested that improved strength and endurance of the back musculature could aid
in the prevention and treatment of LBP [].The spine is a lever subjected to
external loads created by the weight of the trunk and any object lifted, and the
forces created by the various muscles and ligaments surrounding the spine [].
* Winner: hapax-dislegomena-mean, hapax-dislegomena-variance, hapax-dislegomena-
min, sichels-s-mean, sichels-s-variance, sichels-s-min, max-tokens-in-sentence,
max-stop-words-per-sentence, capitalletterwords-words-ratio, capitalletter-
character-ratio, dependency-tree-depth-mean, dependency-tree-depth-max,
* Similar phrase used with Source: or
* Similar phrase used with Source: most common
* The target starts with a definition, but not the source!
^alpha-chars-ratio-mean: 1
!alpha-chars-ratio-variance: 0
^alpha-chars-ratio-min: 1
!honores-h-mean: 0.72
!mean-punctuations-in-sentence: 0.05
!max-punctuations-in-sentence: 0.05
^mean-words-in-sentence: 0.95
```

Figure 8.2: Example of a target snippet presented to the users in the qualitative evaluation. The list of features is added to half of the experiments to help the ranking process.

each target. This was computed using the library "DK-Pro statistics[6]". The results show:

- An inter-rater agreement of 0.299
- An observed disagreement of 0.699
- An expected disagreement of 0.999

We continue to explore the annotator's rank by considering the snippets written by the same author and those written within the same journal (but by different authors). Table 8.3 shows the amount of times users selected the articles in each category of similarity. With a random selection, the expected precision is of 25%.

First, we select a list of stylometric features to extract from the source and the target texts. The literature suggests a broad amount of stylometric features (cf. Mosteller and Wallace 1964; Tweedie and Baayen 1998 or Stamatatos 2009). We filter the content features and some of those who do not make sense in short texts. Table 8.2 presents the list of features we extract for each snippet. In addition, we calculate the minimum, maximum, average and

[6] https://dkpro.github.io/dkpro-statistics/

variance for each of those features across every snippet. We consider the similarity between the source and the targets as a cosine similarity between the feature vectors. Formally, if $V_1 = [v_1; ...; v_n]$ is a vector of the features $v_1; ...; v_n$ and $V_2 = [v_{12}; ...; v_{n2}]$ is a vector of the features $v_{12}; ...; v_{n2}$, their cosine similarity is defined as:

$$similarity = \cos(V_1, V_2) = \frac{\sum_{i=0}^{n} v_i \cdot v_{i2}}{\sum_{i=1}^{n} v_i \cdot \sum_{i=1}^{n} v_{i2}}$$

Snippet/ranking	Most similar(%)	Similar(%)	Less similar(%)	Least similar(%)
Same author	19	45	24	12
Same journal	21	41	26	12

Table 8.1: Pilot study #1: ranking of the crowd-sourcing evaluators for snippets from the same author and snippets from the same journal but different author. The expectation for a random selection is 25%.

As depicted in 8.3, we created box-plots to study whether there is a correlation between the user agreement and the content similarity and in Figure 8.4 one between the user agreement and the writing style similarity.

There is no clear evidence that explains the agreement/disagreement among annotators from the considered features. To dig more deeply, in Figure 8.5 we created a scatter plot to comprehend whether there is a correlation between the three similarities, i.e. content similarity, the writing style similarity and the inter-rater agreement.

The scatter plot does not provide any visual hint about the annotators' agreement/disagreement. In addition, we plotted every combination considering, instead of the whole vector of the aforementioned features (see Table 8.2), each of them singularly. Yet, we did not notice any clear pattern. As there is no additional information added to the previous plot we omitted them in this work.

Finally, we empirically measured whether the annotators did their ranking in a random manner. We executed 500.000 rounds of random studies and, for each of them, calculated the inter-rater agreement using the Krippensdorff's alpha. The results show an average of 0.250 with a variance of 0.020. 28% of the cases have a larger agreement than our study, thus we can conclude

feature name	Description
alpha-chars-ratio	the fraction of total characters in the paragraph which are letters
digit-chars-ratio	the fraction of total characters in the paragraph which are digits
upper-chars-ratio	the fraction of total characters in the paragraph which are upper-case
white-chars-ratio	the fraction of total characters in the paragraph which are whitespace characters
type-token-ratio	ratio between the size of the vocabulary (i.e., the number of different words) and the total number of words
hapax-legomena	the number of words occurring once
hapax-dislegomena	the number of words occurring twice
yules-k	a vocabulary richness measure defined by Yule
simpsons-d	a vocabulary richness measure defined by Simpson
brunets-w	a vocabulary richness measure defined by Brunet
sichels-s	a vocabulary richness measure defined by Sichel
honores-h	a vocabulary richness measure defined by Honore
average-word-length	average length of words in characters
average-sentence-char-length	average length of sentences in characters
average-sentence-word-length	average length of sentences in words

Table 8.2: List of stylometric features used calculate the similarity between text snippets. Many of those features are defined in Tweedie and Baayen, 1998.

with a confidence of 72% that the annotators in our experiment did not rank in a random manner.

8.5 Pilot Study #2

In this study, the evaluators are the authors of the current work. Here, we try to avoid random selection which may be present in the crowd-sourcing platforms. As already mentioned, we showed two different types of experiments to the users. The first type was the same as the one presented to the crowd. In the second, we enrich the presented snippet with a list of features that might help the annotator decide. The first three annotators used the writing style as a feature for ranking the snippets, while the last one used the content information. Table 3 shows for each annotator, the

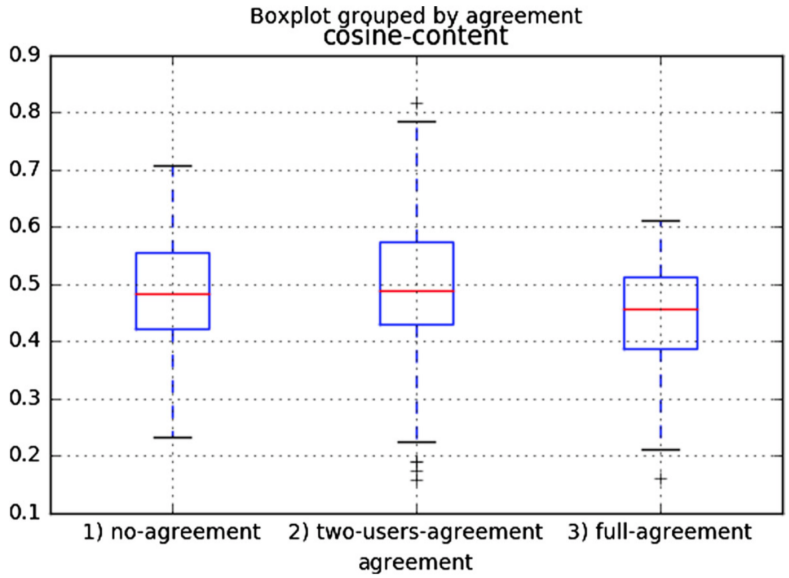


Figure 8.3: Box plots representing the distribution of the annotators' agreement over the similarity of the content.

precision in finding the exact author. The precision for a random selection has an expectation of 25%.

	Precision without features	Precision with features
Annotator #1	0.22	0.22
Annotator #2	0.27	0.29
Annotator #3	0.16	0.25
Annotator #4	0.11	0.15

Table 8.3: Pilot study #2: precision of the annotators in finding the same author as the source snippet. Results are presented for experiments provided with and without features to help for the ranking (the expectation of the precision for a random selection is 25%). Please note: Annotator #4 performed the ranking task wrt. content features

As we can see, the results show a slight improvement in the case the evaluators use the extracted features, but it is not clear if this effect is a result of the additional information presented. To ensure that the content was not key to identifying the correct author, Annotator #4 intentionally conducted

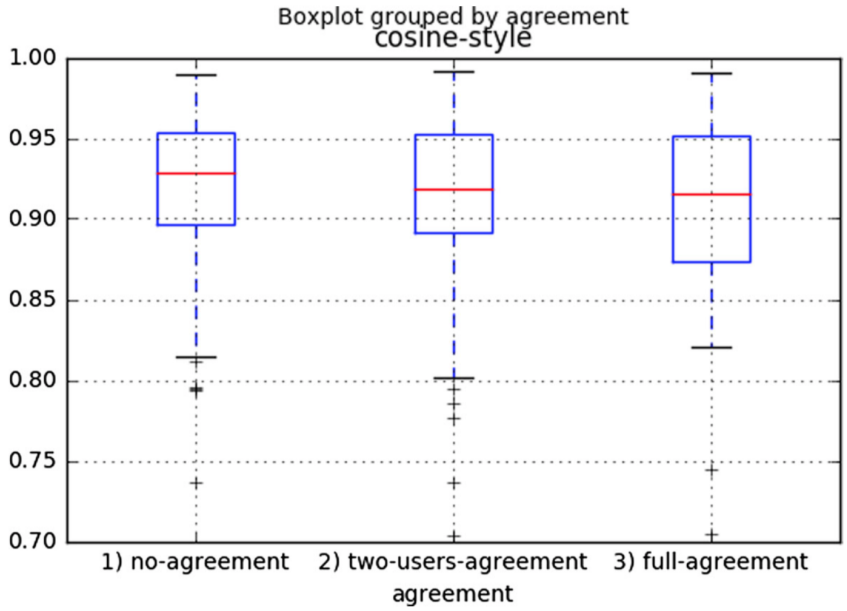


Figure 8.4: Box plots representing the distribution of the annotators' agreement over the similarity of the writing style.

the experiments based on the content resulting in the worst precision values. We collected the observation from each of the annotators and present them below. These can serve either as confirmation of existing features, or as inspirations for new features for automatic authorship identification tasks.

Annotator #1 The first observation, which is specific to the data set is the initial sentence of the paragraph. Often authors tend to give a definition of the key terms of the text, for example the sentence may start with "Breast cancer is the most common cancer [...]". Other authors tend to use more active voice to raise the awareness via phrases like "Break cancer remains the most common cancer". The last example also represents a writing style which makes use of temporal aspects, which includes phrases like "since", "more recently", "recent advances in [...]", or specific dates like "in 2013".

Another observation that might be specific to the data set is the varying degree of granularity for named entities, for example "Liver cancer", in

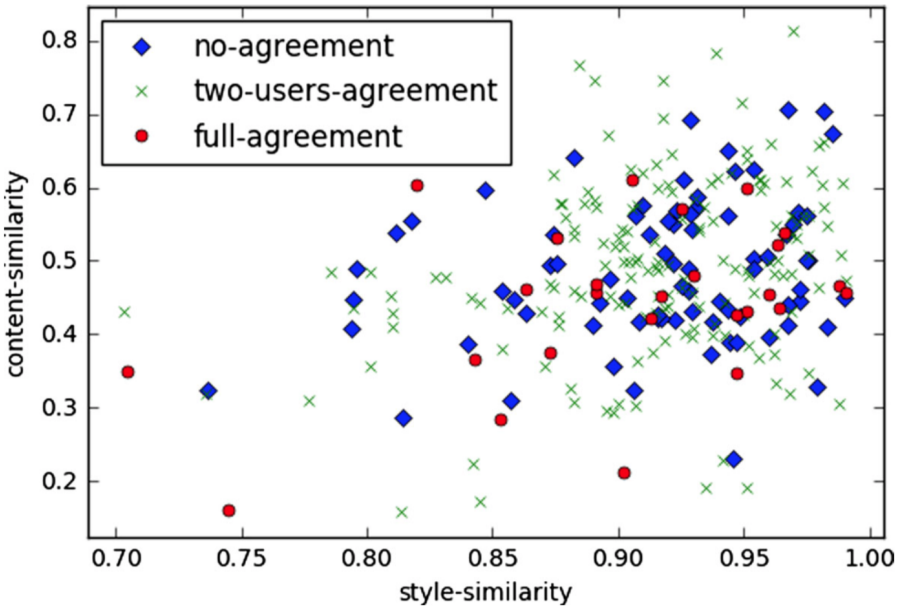


Figure 8.5: Scatter plot relating three dimensions: style similarity, content similarity and agreement between annotators.

contrast to "Primary liver cancer". Related to this is the spelling of certain entities, for example one author may write "Castelman's disease", while others may refer to the same concept as "Castelman disease". The same also applies to the capitalization of concept, for example "short linear motifs", or "Short Linear Motifs". To a lesser extend, this has also been observed for abbreviations ("HIV1", "hiv", "HIV-1").

Specifically in scientific literature, there are only a few authors that make use of words that are considered too emotional or imprecise, for example "tremendous" or "dramatic". The level of preciseness also varies between authors, some may state "approximately 500,000 cases", while others try to be more exact, for example "528,000 cases". Another potential indicator for specific authors is the use of "slang abbreviations", for example "can't" instead of "cannot". Given that authors are consistent and use the same spelling for all their written text, different authors could be distinguished. This also applies to synonyms ("high blood pressure" vs. "hypertension") or alternative spellings ("etiology" vs. "aetiology").

The initial words of a sentence often appear to convey information about the specific writing style. Some authors use phrases to link two sentences. Examples are: “Indeed”, “Although”, “Hence”, “Thus”, “While”, “Moreover” and many more. The usage of such words or just specific words might be indicative of certain authors. Furthermore, the presence (or absence) of a comma following such words, might be a useful feature to distinguish two writing styles. Some authors also appear to have the initial words to indicate a temporal aspect, for example “Since then”.

Trigger phrases may also occur within the sentence, which are characteristic for specific writing style, for example: “not very long ago”. A special case of such phrase is “comprise”, where some authors add the word “of”. This might also be indicative for the distinction between native and non-native speakers. Native speakers may also tend to use words, which are less frequent. Another obvious difference is whether the author uses British or American spelling, which might also be due to the intended target audience. The way numbers are written might also be specific to the locale of the text in addition to the preference of the author, for example “55,000”, “55.000” or “55’000”. Similar to this feature is the usage of the “Oxford comma”, where a comma is also used for the last item in an enumeration (“foo, bar, and baz” vs. “foo, bar and baz”); its presence might help to separate two authors. In general, the usage of semicolons, dashes and brackets might also be specific to some authors.

Another feature, which requires a certain level of consistency is the usage of multi-word terms, which might be written as single word, separated by a hyphen or written as separate words. Examples for this observation are: “US-born” versus “US born”, “over weight” versus “overweight” and “crossing-over” versus “crossing over”.

Annotator #2 After getting an overview of the first example snippets, it appeared that the informative value of typical features like the comma usage, sentence length or usage of brackets was not sufficient to arrive at a conclusion. In particular when the style is, in some cases, dependent upon the publisher, which is typically the case in scientific literature (e.g. citation style). In general, it seemed that stylistic choices and preferences of the author had a bigger impact. And sometimes the distinct absence of particular

styles was more informative. Some authors appear to favor including precise figures within their text while others tend to use descriptive language (e.g. "about one-third", "30%", "30.2%"). In general, the usage, or their absence, of precise figures within the text seems to be a highly informative characteristic of the authors. Furthermore, some authors seem to have the tendency to list particular information, separated by commas, within their text, while others do not use this style at all. Another distinctive feature was the usage of abbreviations. First, some authors used a lot of abbreviations while others did not use them at all. Second, some defined these abbreviations and did not reuse them later in the text, although it would have been appropriate.

Annotator #3 The identification of the similarity in the writing style starts by analyzing the structure of the sentences for each snippet. The formulation of the first sentence is the main observed characteristic judged in a snippet. Authors expressing the same concepts once as a subject and once as an object leads to considering the texts as written by different authors. For example, "Both obesity and metabolic syndrome (MetS) are well known." and "A major obstacle in the treatment of overweight and obesity is hunger." express "obesity" as either the subject or the object.

In the cases where the first sentence starts similarly in various target snippets, the way it is expressed also provide a differentiating feature. Some authors favor details compared to the short sentence. Furthermore, the distinction extends to the structure of the snippet. Some authors tend to favor long sentences to short ones. The structure of these sentences is seen as a characteristic that helps to decide the similarity between the presented texts.

Other aspects that were used to judge were:

- The way numeric values are represented (some use a comma or dots: 400.000 vs. 400,000).
- Whether there was a different representation of statistics (some favor percentage to total amounts: example: 350.000 vs. 10% of female).
- The use of enumeration (some authors like to enumerate information, and some prefer to have a description for each of the details).

Also, the specific phrases used, gave hints about the authorship attribution. For example, phrases like, "which" or "although" were considered very informative.

Annotator #4 For the sake of comparison between rankings based on writing style versus content, we add the ranking approach of Annotator #4:

After the first ranking task, it became apparent that an understanding of the source's content was essential to conduct the ranking. Similar to a summarization task, factual information was identified. This identification procedure can be best described as identi- fying answers to the 5W question types, i.e. who, what, when, where and how. It could be observed that the temporal dimension, for instance, the year the source's facts were referring to, was helpful when ranking the targets. Comparing the facts from the source with the ones to be ranked turned out as the common procedure for the ranking; for this part the support of domain experts familiar with synonymous medical expressions would have sometimes been beneficial.

As a first step, targets which were off-topic or out-of-domain were placed at the bottom of the ranking. Then, a closer look was taken at the targets sharing most of the facts with the source. These targets were ranked according to the coverage of content, i.e. the number of shared facts. In some experiments for example, the targets were missing facts about a certain region a disease was found. In several cases, two targets shared an equal number of facts with the source; yet one of them offered more information, i.e. more facts than the other one. These cases were considered as less similar to the source as well. In cases where the amount of facts were more or less equal, the following two indicators influenced the ranking. First, the order in which the information is presented—same order as in the source equalled a higher similarity than a different order. Second, if the order of information was the same, readability tipped the scales in favor of the more readable than the less readable one.

8.6 Conclusion and Future Work

In this work, we have conducted two extensive studies to gain a better understanding (1) how humans judge an author's writing style (being faced with text passages exhibiting a high-content similarity) and (2) which content-agnostic, stylometric features they preferably use to identify an author. These experiences and observations contribute to automate (mimick) this process by identifying as well as by distinguishing specific features used by humans in their decision making process. We provide detailed descriptions and observations of this author identification process which will prove valuable in an effort to develop algorithms, for instance, in areas like plagiarism detection or forensic analyses. Furthermore, our findings indicate that the task turns out to be very challenging, especially with the current experiment settings. The results also indicate that the annotation process from the crowd is more likely not to have random evaluations (72% confidence). In addition, we have made our data set publicly available to the research community to enable further investigations and algorithm development.

In future work, we plan to take the author's institution into account to serve as a dimension in the paper selection process—similarly with papers published in the same journal. We also plan to take into account text passages written within a distance of years by the same author. We plan to extend our study by increasing and diversifying the set of experiments aiming to capture, from human annotators, properties of the thinking process while performing this task. We also intend to automatically learn and suggest personalized features for each of the annotators, helping them rank, according to their metrics.

Bibliography

Abacha, Asma Ben and Pierre Zweigenbaum (2011). "Medical entity recognition: A comparison of semantic and statistical methods." In: *Proceedings of BioNLP 2011 Workshop*. Association for Computational Linguistics, pp. 56–64 (cit. on p. 151).

Bergsma, Shane, Matt Post, and David Yarowsky (2012). "Stylometric analysis of scientific articles." In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 327–337 (cit. on p. 152).

Brocardo, Marcelo Luiz et al. (2013). "Authorship verification for short messages using stylometry." In: *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, pp. 1–6 (cit. on p. 151).

Corney, David PA et al. (2004). "BioRAT: extracting biological information from full-length papers." In: *Bioinformatics* 20.17, pp. 3206–3213 (cit. on p. 151).

Holmes, David I (1998). "The evolution of stylometry in humanities scholarship." In: *Literary and linguistic computing* 13.3, pp. 111–117 (cit. on pp. 150, 152).

Inches, Giacomo, Morgan Harvey, and Fabio Crestani (2013). "Finding participants in a chat: Authorship attribution for conversational documents." In: *2013 International Conference on Social Computing*. IEEE, pp. 272–279 (cit. on p. 151).

Iqbal, Farkhund et al. (2013). "A unified data mining solution for authorship analysis in anonymous textual communications." In: *Information Sciences* 231, pp. 98–112 (cit. on p. 151).

Juola, Patrick et al. (2008). "Authorship attribution." In: *Foundations and Trends® in Information Retrieval* 1.3, pp. 233–334 (cit. on p. 150).

Kern, Roman et al. (2012). "Teambeam-meta-data extraction from scientific literature." In: *D-Lib Magazine* 18.7, p. 1 (cit. on p. 152).

Knaap, L and FA Grootjen (2007). "Author identification in chatlogs using formal concept analysis." In: (cit. on p. 151).

Krippendorff, Klaus (2004). "Content analysis: An introduction to its methodology Thousand Oaks." In: *Calif.: Sage* (cit. on p. 156).

Liakata, Maria et al. (2012). "Automatic recognition of conceptualization zones in scientific articles and two life science applications." In: *Bioinformatics* 28.7, pp. 991–1000 (cit. on p. 151).

Mendenhall, Thomas Corwin (1887). "The characteristic curves of composition." In: *Science* 9.214, pp. 237–249 (cit. on p. 150).

Mosteller, Frederick and David Wallace (1964). *Inference and disputed authorship: The Federalist.(1964)* (cit. on p. 150).

Ravenscroft, James, Maria Liakata, and Amanda Clare (2013). "Partridge: An effective system for the automatic cassification of the types of academic papers." In: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, pp. 351–358 (cit. on p. 152).

Rexha, Andi et al. (2015). "Towards Authorship Attribution for Bibliometrics using Stylometric Features." In: *CLBib@ ISSI*, pp. 44–49 (cit. on pp. 148, 150).

Rexha, Andi et al. (2016). "Towards a More Fine Grained Analysis of Scientific Authorship: Predicting the Number of Authors Using Stylometric Features." In: *BIR@ ECIR*, pp. 26–31 (cit. on pp. 148, 150).

Silva, Rui Sousa et al. (2011). "'twazn me!!!;('automatic authorship analysis of micro-blogging messages." In: *International Conference on Application of Natural Language to Information Systems*. Springer, pp. 161–168 (cit. on p. 151).

Stamatatos, Efstathios (2009). "A survey of modern authorship attribution methods." In: *Journal of the American Society for information Science and Technology* 60.3, pp. 538–556 (cit. on pp. 150, 151).

Tsuruoka, Yoshimasa, Junichi Tsujii, and Sophia Ananiadou (2008). "FACTA: a text search engine for finding associated biomedical concepts." In: *Bioinformatics* 24.21, pp. 2559–2560 (cit. on p. 151).

Tweedie, Fiona J and R Harald Baayen (1998). "How variable may a constant be? Measures of lexical richness in perspective." In: *Computers and the Humanities* 32.5, pp. 323–352 (cit. on p. 159).

Villar-Rodriguez, Esther et al. (2016). "A feature selection method for author identification in interactive communications based on supervised learning and language typicality." In: *Engineering Applications of Artificial Intelligence* 56, pp. 175–184 (cit. on p. 151).

Zhang, Chunxia et al. (2014). "Authorship identification from unstructured texts." In: *Knowledge-Based Systems* 66, pp. 99–111 (cit. on p. 151).

Bibliography

Abulaish, Muhammad and Amit Kumar Sah (2019). "A Text Data Augmentation Approach for Improving the Performance of CNN." In: *11th International Conference on Communication Systems & Networks, COMSNETS 2019, Bengaluru, India, January 7-11, 2019*. IEEE, pp. 625–630. DOI: 10.1109/COMSNETS.2019.8711054. URL: https://doi.org/10.1109/COMSNETS.2019.8711054 (cit. on p. 17).

Agarwal, Apoorv, Fadi Biadsy, and Kathleen R. Mckeown (2009). "Contextual Phrase-level Polarity Analysis Using Lexical Affect Scoring and Syntactic N-grams." In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '09 (cit. on p. 15).

Akbik, Alan and Alexander Löser (2012). "KrakeN: N-ary Facts in Open Information Extraction." In: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. AKBC-WEKEX '12. Montreal, Canada: Association for Computational Linguistics, pp. 52–56. URL: http://dl.acm.org/citation.cfm?id=2391200.2391210 (cit. on p. 7).

Alon, Uri, Omer Levy, and Eran Yahav (2018). "code2seq: Generating Sequences from Structured Representations of Code." In: *CoRR* abs/1808.01400. arXiv: 1808.01400. URL: http://arxiv.org/abs/1808.01400 (cit. on p. 11).

Barbosa, Luciano and Junlan Feng (2010). "Robust Sentiment Detection on Twitter from Biased and Noisy Data." In: *COLING (Posters)*, pp. 36–44 (cit. on pp. 8, 15).

Bast, Hannah and Elmar Haussmann (2013). "Open Information Extraction via Contextual Sentence Decomposition." In: *Proceedings of the 2013 IEEE 7th International Conference on Semantic Computing*. ICSC '13. IEEE. Irvine, CA, USA (cit. on p. 7).

Baxendale, Phyllis B (1958). "Machine-made index for technical literature—an experiment." In: *IBM Journal of research and development* 2.4, pp. 354–361 (cit. on p. 12).

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation." In: *Journal of machine Learning research* 3.Jan, pp. 993–1022 (cit. on p. 14).

Brocardo, Marcelo Luiz et al. (2013). "Authorship verification for short messages using stylometry." In: *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, pp. 1–6 (cit. on p. 20).

Cambria, Erik (2016). "Affective Computing and Sentiment Analysis." In: *IEEE Intelligent Systems* 31.2, pp. 102–107. DOI: 10.1109/MIS.2016.31. URL: http://dx.doi.org/10.1109/MIS.2016.31 (cit. on p. 8).

Cao, Ziqiang et al. (2015). "Ranking with recursive neural networks and its application to multi-document summarization." In: *Twenty-ninth AAAI conference on artificial intelligence* (cit. on p. 11).

Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." In: *arXiv preprint arXiv:1406.1078* (cit. on p. 17).

Clarizia, Fabio et al. (2011). "A new text classification technique using small training sets." In: *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011, Córdoba, Spain, November 22-24, 2011*. Ed. by Sebastián Ventura et al. IEEE, pp. 1038–1043. DOI: 10.1109/ISDA.2011.6121795. URL: https://doi.org/10.1109/ISDA.2011.6121795 (cit. on p. 17).

Cui, Lei, Furu Wei, and Ming Zhou (2018). "Neural open information extraction." In: *arXiv preprint arXiv:1805.04270* (cit. on p. 7).

Culwin, Fintan and Thomas Lancaster (2001). "Plagiarism, prevention, deterrence and detection." In: *Available for ILT members from* (cit. on p. 20).

Deerwester, Scott et al. (1990). "Indexing by latent semantic analysis." In: *Journal of the American society for information science* 41.6, pp. 391–407 (cit. on p. 14).

Del Corro, Luciano and Rainer Gemulla (2013). "ClausIE: Clause-based Open Information Extraction." In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, pp. 355–366. ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488420. URL: http://doi.acm.org/10.1145/2488388.2488420 (cit. on p. 7).

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423 (cit. on p. 14).

Elekes, Ábel et al. (2019). "Learning from Few Samples: Lexical Substitution with Word Embeddings for Short Text Classification." In: *19th ACM/IEEE Joint Conference on Digital Libraries, JCDL 2019, Champaign, IL, USA, June 2-6, 2019*. Ed. by Maria Bonn et al. IEEE, pp. 111–119. DOI: 10.1109/JCDL.2019.00025. URL: https://doi.org/10.1109/JCDL.2019.00025 (cit. on p. 18).

Erkan, Günes and Dragomir R Radev (2004). "Lexrank: Graph-based lexical centrality as salience in text summarization." In: *Journal of artificial intelligence research* 22, pp. 457–479 (cit. on p. 11).

Fader, Anthony, Stephen Soderland, and Oren Etzioni (2011). "Identifying Relations for Open Information Extraction." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 1535–1545. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145596 (cit. on p. 7).

Fellbaum, Christiane (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press (cit. on p. 13).

Fernandes, Patrick, Miltiadis Allamanis, and Marc Brockschmidt (2018). "Structured Neural Summarization." In: *CoRR* abs/1811.01824. arXiv: 1811.01824. URL: http://arxiv.org/abs/1811.01824 (cit. on p. 11).

Gillick, Dan and Benoit Favre (2009). "A scalable global model for summarization." In: *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pp. 10–18 (cit. on p. 11).

Go, Alec, Richa Bhayani, and Lei Huang (2009). *Twitter Sentiment Classification using Distant Supervision*. CS224N Project Report, Standford University (cit. on pp. 8, 15).

He, Ruidan et al. (2018). "Exploiting Document Knowledge for Aspect-level Sentiment Classification." In: *CoRR* abs/1806.04346. arXiv: 1806.04346. URL: http://arxiv.org/abs/1806.04346 (cit. on p. 9).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory." In: *Neural computation* 9.8, pp. 1735–1780 (cit. on p. 17).

Hofmann, Thomas (2013). "Probabilistic latent semantic analysis." In: *arXiv preprint arXiv:1301.6705* (cit. on p. 14).

Holmes, David I (1998). "The evolution of stylometry in humanities scholarship." In: *Literary and linguistic computing* 13.3, pp. 111–117 (cit. on p. 19).

Hu, Minqing and Bing Liu (2004). "Mining and summarizing customer reviews." In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*. Ed. by Won Kim et al. ACM, pp. 168–177. DOI: 10.1145/1014052.1014073. URL: http://doi.acm.org/10.1145/1014052.1014073 (cit. on p. 8).

Jakob, Niklas and Iryna Gurevych (2010). "Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields." In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pp. 1035–1045. URL: http://www.aclweb.org/anthology/D10-1101 (cit. on p. 8).

Jin, Wei, Hung Hay Ho, and Rohini K. Srihari (2009). "OpinionMiner: a novel machine learning system for web opinion mining and extraction." In: *KDD*, pp. 1195–1204 (cit. on p. 8).

Joulin, Armand et al. (2016). "Bag of Tricks for Efficient Text Classification." In: *arXiv preprint arXiv:1607.01759* (cit. on p. 14).

Juola, Patrick et al. (2008). "Authorship attribution." In: *Foundations and Trends® in Information Retrieval* 1.3, pp. 233–334 (cit. on p. 20).

Kern, Roman and Michael Granitzer (2009). "Efficient linear text segmentation based on information retrieval techniques." In: *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pp. 167–171 (cit. on p. 21).

Kim, Byungsoo, Hwanjo Yu, and Gary Geunbae Lee (2016). "Automatic Open Knowledge Acquisition via Long Short-Term Memory Networks with Feedback Negative Sampling." In: *arXiv preprint arXiv:1605.07918* (cit. on p. 7).

Kim, Kang-Min et al. (2019). "From Small-scale to Large-scale Text Classification." In: *The World Wide Web Conference, WWW 2019, San Francisco,*

CA, USA, May 13-17, 2019. Ed. by Ling Liu et al. ACM, pp. 853–862. DOI: 10.1145/3308558.3313563. URL: https://doi.org/10.1145/3308558.3313563 (cit. on p. 17).

Klampfl, Stefan et al. (2014). "Unsupervised document structure analysis of digital scientific articles." In: *International journal on digital libraries* 14.3-4, pp. 83–99 (cit. on p. 21).

Knight, Kevin and Daniel Marcu (2002). "Summarization beyond sentence extraction: A probabilistic approach to sentence compression." In: *Artificial Intelligence* 139.1, pp. 91–107 (cit. on p. 11).

Landauer, Thomas K, Peter W Foltz, and Darrell Laham (1998). "An introduction to latent semantic analysis." In: *Discourse processes* 25.2-3, pp. 259–284 (cit. on p. 14).

Lewis, Mike et al. (2019). "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." In: *arXiv preprint arXiv:1910.13461* (cit. on p. 11).

Li, Fangtao, Minlie Huang, and Xiaoyan Zhu (2010). "Sentiment Analysis with Global Topics and Local Dependency." In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1913 (cit. on p. 9).

Liu, Bing, Minqing Hu, and Junsheng Cheng (2005). "Opinion observer: analyzing and comparing opinions on the Web." In: *WWW*, pp. 342–351 (cit. on p. 8).

Liu, Fei et al. (2018). "Toward abstractive summarization using semantic representations." In: *arXiv preprint arXiv:1805.10399* (cit. on p. 11).

Liu, Y et al. (2019). "RoBERTa: A robustly optimized BERT pretraining approach. arXiv 2019." In: *arXiv preprint arXiv:1907.11692* (cit. on p. 14).

Lochter, Johannes V. et al. (2018). "Semantic Indexing-Based Data Augmentation for Filtering Undesired Short Text Messages." In: *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018*. Ed. by M. Arif Wani et al. IEEE, pp. 1034–1039. DOI: 10.1109/ICMLA.2018.00169. URL: https://doi.org/10.1109/ICMLA.2018.00169 (cit. on p. 17).

Lu, Xinghua et al. (2006). "Research Paper: Enhancing Text Categorization with Semantic-enriched Representation and Training Data Augmentation." In: *JAMIA* 13.5, pp. 526–535. DOI: 10.1197/jamia.M2051. URL: https://doi.org/10.1197/jamia.M2051 (cit. on p. 17).

Mausam et al. (2012). "Open Language Learning for Information Extraction." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 523–534. URL: http://dl.acm.org/citation.cfm?id=2390948.2391009 (cit. on p. 7).

Mei, Qiaozhu et al. (2007). "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs." In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: ACM, pp. 171–180. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242596. URL: http://doi.acm.org/10.1145/1242572.1242596 (cit. on p. 9).

Mendenhall, Thomas Corwin (1887). "The characteristic curves of composition." In: *Science* 9.214, pp. 237–249 (cit. on p. 19).

Mihalcea, Rada and Paul Tarau (2004). "Textrank: Bringing order into text." In: *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411 (cit. on p. 11).

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781* (cit. on p. 14).

Mosteller, Frederick and David Wallace (1964). *Inference and disputed authorship: The Federalist.(1964)* (cit. on p. 19).

Mukherjee, Arjun and Bing Liu (2012). "Aspect Extraction Through Semi-supervised Modeling." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. ACL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 339–348. URL: http://dl.acm.org/citation.cfm?id=2390524.2390572 (cit. on p. 9).

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." In: *Proceedings of EMNLP*. Philadelphia: Association for Computational Linguistics, pp. 79–86 (cit. on p. 8).

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "Glove: Global vectors for word representation." In: *In EMNLP* (cit. on p. 14).

Peters, Matthew E. et al. (2018). "Deep contextualized word representations." In: *Proc. of NAACL* (cit. on p. 14).

Raffel, Colin et al. (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer." In: *Journal of Machine Learning Research* 21.140, pp. 1–67 (cit. on p. 11).

Rizos, Georgios, Konstantin Hemker, and Björn W. Schuller (2019). "Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification." In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. Ed. by Wenwu Zhu et al. ACM, pp. 991–1000. DOI: 10.1145/3357384.3358040. URL: https://doi.org/10.1145/3357384.3358040 (cit. on p. 17).

Schulz, Sarah et al. (2016). "Multimodular Text Normalization of Dutch User-Generated Content." In: *ACM TIST* 7.4, 61:1–61:22. DOI: 10.1145/2850422. URL: https://doi.org/10.1145/2850422 (cit. on p. 17).

Seifert, Christin et al. (2013). "Text Representation for Efficient Document Annotation." In: *J. UCS* 19.3, pp. 383–405 (cit. on p. 11).

Shridhar, Kumar et al. (2019). "Subword Semantic Hashing for Intent Classification on Small Datasets." In: *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*. IEEE, pp. 1–6. DOI: 10.1109/IJCNN.2019.8852420. URL: https://doi.org/10.1109/IJCNN.2019.8852420 (cit. on p. 17).

Stamatatos, Efstathios (2009). "A survey of modern authorship attribution methods." In: *Journal of the American Society for information Science and Technology* 60.3, pp. 538–556 (cit. on p. 20).

Stein, Benno, Moshe Koppel, and Efstathios Stamatatos (2007). "Plagiarism analysis, authorship identification, and near-duplicate detection PAN'07." In: *ACM SIGIR Forum*. Vol. 41. 2. ACM New York, NY, USA, pp. 68–71 (cit. on p. 20).

Titov, Ivan and Ryan McDonald (2008). "A Joint Model of Text and Aspect Ratings for Sentiment Summarization." In: *PROC. ACL-08: HLT*. Pp. 308–316 (cit. on p. 9).

Toutanova, Kristina et al. (2001). "Text Classification in a Hierarchical Mixture Model for Small Training Sets." In: *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*. ACM, pp. 105–112. DOI: 10.1145/502585.502604. URL: https://doi.org/10.1145/502585.502604 (cit. on p. 16).

Veliz, Claudia Matos, Orphée De Clercq, and Véronique Hoste (2019). "Benefits of Data Augmentation for NMT-based Text Normalization of User-Generated Content." In: *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019, Hong Kong, China, November 4, 2019*. Ed. by Wei Xu et al. Association for Computational Linguistics, pp. 275–285. DOI: 10.18653/v1/D19-5536. URL: https://doi.org/10.18653/v1/D19-5536 (cit. on p. 17).

Villar-Rodriguez, Esther et al. (2016). "A feature selection method for author identification in interactive communications based on supervised learning and language typicality." In: *Engineering Applications of Artificial Intelligence* 56, pp. 175–184 (cit. on p. 20).

Wang, Sida I. and Christopher D. Manning (2012). "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification." In: *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*. The Association for Computer Linguistics, pp. 90–94. URL: https://www.aclweb.org/anthology/P12-2018/ (cit. on p. 18).

Wang, Wenya et al. (2016). "Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 616–626. DOI: 10.18653/v1/D16-1059. URL: https://www.aclweb.org/anthology/D16-1059 (cit. on p. 9).

Wei, Jason W. and Kai Zou (2019). "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui et al. Association for Computational Linguistics, pp. 6381–6387. DOI: 10.18653/v1/D19-1670. URL: https://doi.org/10.18653/v1/D19-1670 (cit. on p. 17).

Wu, Fei and Daniel S. Weld (2010). "Open Information Extraction Using Wikipedia." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 118–127. URL: http://dl.acm.org/citation.cfm?id=1858681.1858694 (cit. on p. 7).

Wu, Yuanbin et al. (2009). "Phrase Dependency Parsing for Opinion Mining." In: *Proceedings of the 2009 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp. 1533–1541. URL: http://www.aclweb.org/anthology/D09-1159 (cit. on p. 9).

Xu, Hu et al. (2018). "Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 592–598. DOI: 10.18653/v1/P18-2094. URL: https://www.aclweb.org/anthology/P18-2094 (cit. on p. 9).

Yates, Alexander et al. (2007). "TextRunner: Open Information Extraction on the Web." In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. NAACL-Demonstrations '07. Rochester, New York: Association for Computational Linguistics, pp. 25–26. URL: http://dl.acm.org/citation.cfm?id=1614164.1614177 (cit. on pp. 5, 7).

Yu, Shujuan et al. (2019). "Hierarchical Data Augmentation and the Application in Text Classification." In: *IEEE Access* 7, pp. 185476–185485. DOI: 10.1109/ACCESS.2019.2960263. URL: https://doi.org/10.1109/ACCESS.2019.2960263 (cit. on p. 17).

Zechner, Mario et al. (2009). "External and intrinsic plagiarism detection using vector space models." In: *Proc. SEPLN*. Vol. 32, pp. 47–55 (cit. on p. 20).

Zhang, Chunxia et al. (2014). "Authorship identification from unstructured texts." In: *Knowledge-Based Systems* 66, pp. 99–111 (cit. on p. 20).