EACL 2017

European Chapter of the Association for Computational Linguistics

Proceedings of the Student Research Workshop

April 3-7, 2017

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-945626-37-1

Introduction

Welcome to the EACL 2017 Student Research Workshop (SRW).

Following previous Student Research Workshops, this year we have two different kinds of submissions: research papers and thesis proposals. Thesis proposals are intended for advanced students who have decided on a thesis topic and wish to get feedback on their proposal and broader ideas for their continuing work, while research papers can describe completed work or work in progress with preliminary results.

We received 10 thesis proposals and 25 research papers this year. Out of these, we accepted 2 thesis proposals and 10 research papers, leading to an acceptance rate of 20% for thesis proposals and 40% for research papers.

All the papers will be presented at the main conference poster session, giving the opportunity for students to interact and present their work to a large and diverse audience.

In addition to this, each accepted SRW paper is assigned a dedicated mentor. The mentor is an experienced researcher from academia or industry who will prepare in-depth comments and questions in advance of the poster session and will provide feedback to the student author.

This year, EACL awarded travel grants to 13 students who were presenting their work either in the SRW or in the main conference. We received a total of 27 applications for studentships, meaning that 48% of them were accepted after a competitive selection process.

We are very grateful to our program committee members who gave constructive and detailed reviews for each of the student papers. We would also like to acknowledge researchers who agreed to mentor and provide expert feedback on the student papers. Many thanks to our faculty adviser Barbara Plank for her invaluable guidance, as well as the EACL 2017 organizing committee for their constant support and suggestions. Finally, we thank all students for their submissions and participation in the SRW.

John J. Camilleri
Mariona Coll Ardanuy
Uxoa Iñurrieta
Florian Kunneman

Organizers:

John J. Camilleri (University of Gothenburg)
Mariona Coll Ardanuy (University of Göttingen)
Uxoa Iñurrieta (University of the Basque Country)
Florian Kunneman (Radboud University)

Faculty advisor:

Barbara Plank (University of Groningen)

Mentoring:

Chloé Braud (University of Copenhagen)
Grzegorz Chrupala (University of Tilburg)
Tommaso Fornaciari (Università di Trento)
Alexander Gelbukh (Instituto Politécnico Nacional)
Ivan Habernal (Technische Universität Darmstadt)
Claudia Hauff (Delft University of Technology)
Dirk Hovy (University of Copenhagen)
Maarten Marx (Universiteit van Amsterdam)
Vincent Ng (University of Texas at Dallas)
Ted Pedersen (University of Minnesota)
Massimo Poesio (University of Essex)
Anders Soegaard (University of Copenhagen)

Program Committee:

Željko Agić (IT University Copenhagen)
Héctor Martínez Alonso (INRIA - Paris 7)
Yoav Artzi (Cornell University)
Yonatan Bisk (University of Illinois Urbana-Champaign)
Gosse Bouma (University of Groningen)
António Branco (University of Lisbon)
Shu Cai (University of Southern California)
Monojit Choudhury (Microsoft Research India)
Orphee Declerq (University College Ghent)
Simon Dobnik (University of Gothenburg)
Marieke van Erp (VU University Amsterdam)
Kilian Evang (University of Groningen)
Thomas François (UC Louvain)

Michael Gamon (Microsoft)
Alexander Gelbukh (Instituto Politécnico Nacional)
Debanjan Ghosh (Rutgers University)
Kevin Gimpel (Toyota Technological Institute at Chicago)
Mena Habib (Maastricht University)
Eva Hasler (University of Cambridge)
Claudia Hauff (Delft University of Technology)
Lars Hellan (Norwegian University of Science and Technology)
Iris Hendrickx (Radboud University)
Dirk Hovy (University of Copenhagen)
Anders Johannsen (Apple)
Richard Johannsen (University of Gothenburg)
Ehsan Khoddammohammadi (University of Amsterdam)
Philipp Koehn (Johns Hopkins University (DOUBLE))
Daniel de Kok (University of Tübingen)
Varada Kolhatkar (University of Toronto)
Yannis Korkontzelos (Edge Hill University)
Staffan Larsson (University of Gothenburg)
Lori Levin (Carnegie Mellon University)
Junyi Jessy Li (University of Pennsylvania)
Peter Ljunglöf (Chalmers University of Technology)
Shervin Malmasi (Harvard Medical School)
Daniel Marcu (University of Southern California)
Thomas Meyer (Google Zurich)
Taesun Moon (IBM Research)
Alessandro Moschitti (Qatar Computing Research Institute)
Graham Neubig (Carnegie Mellon University)
Vincent Ng (University of Texas at Dallas)
Kemal Oflazer (Carnegie Mellon University Qatar)
Yannick Parmentier (University of Orléans)
Ted Pedersen (University of Minnesota Duluth)
Nanyun Peng (Johns Hopkins University)
Preethi Raghavan (IBM TJ Watson Research Center)
Robert Remus (University of Leipzig)
Michael Roth (University of Edinburgh)
Satoshi Sekine (New York University)
Kairit Sirts (Tallinn University of Technology)
Swapna Somasundaran (ETS)
Richard Sproat (Google)
Keh-Yih Su (Academia Sinica)
Christoph Teichmann (University of Potsdam)
Jesús Calvillo Tinoco (University of Saarland)
Eva Maria Vecchi (University of Cambridge)
Suzan Verberne (Radboud University)
Nina Wacholder (Rutgers University)
William Yang Wang (Carnegie Mellon University)

Bonnie Webber (University of Edinburgh)
Travis Wolfe (John Hopkins University)
Bishan Yang (Carnegie Mellon University)
Meishan Zhang (Singapore University of Technology and Design)
Yuan Zhang (Massachusetts Institute of Technology)
Yue Zhang (Singapore University of Technology and Design)

Conference Program

17:30–19:30 *Pragmatic descriptions of perceptual stimuli*
Emiel van Miltenburg

17:30–19:30 *Replication issues in syntax-based aspect extraction for opinion mining*
Edison Marrese-Taylor and Yutaka Matsuo

17:30–19:30 *Discourse Relations and Conjoined VPs: Automated Sense Recognition*
Valentina Pyatkin and Bonnie Webber

17:30–19:30 *Deception detection in Russian texts*
Olga Litvinova, Pavel Seredin, Tatiana Litvinova and John Lyell

17:30–19:30 *A Computational Model of Human Preferences for Pronoun Resolution*
Olga Seminck and Pascal Amsili

17:30–19:30 *Automatic Extraction of News Values from Headline Text*
Alicja Piotrkowicz, Vania Dimitrova and Katja Markert

17:30–19:30 *Assessing Convincingness of Arguments in Online Debates with Limited Number of Features*
Lisa Andreevna Chalaguine and Claudia Schulz

17:30–19:30 *Zipf's and Benford's laws in Twitter hashtags*
José Alberto Pérez-Melián, J. Alberto Conejero and Cesar Ferri Ramírez

17:30–19:30 *A Multi-aspect Analysis of Automatic Essay Scoring for Brazilian Portuguese*
Evelin Amorim and Adriano Veloso

17:30–19:30 *Detecting spelling variants in non-standard texts*
Fabian Barteld

17:30–19:30 *Literal or idiomatic? Identifying the reading of single occurrences of German multiword expressions using word embeddings*
Rafael Ehren

17:30–19:30 *Evaluating the Reliability and Interaction of Recursively Used Feature Classes for Terminology Extraction*
Anna Hätty, Michael Dorna and Sabine Schulte im Walde

Pragmatic descriptions of perceptual stimuli

Emiel van Miltenburg
Vrije Universiteit Amsterdam
emiel.van.miltenburg@vu.nl

Abstract

This research proposal discusses pragmatic factors in image description, arguing that current automatic image description systems do not take these factors into account. I present a general model of the human image description process, and propose to study this process using corpus analysis, experiments, and computational modeling. This will lead to a better characterization of human image description behavior, providing a road map for future research in automatic image description, and the automatic description of perceptual stimuli in general.

1 Introduction

Automatic image description is a key challenge at the intersection of Computer Vision (CV) and Natural Language Processing (NLP), because it requires a deep understanding of both images and natural language (Bernardi et al., 2016). There are two major datasets that are used to train and evaluate automatic image description models: Flickr30K (Young et al. (2014); 30K images) and MS COCO (Lin et al. (2014); 150K images). These descriptions were collected through a crowdsourcing task where Workers were asked to provide one-sentence descriptions for each image. One of the assumptions behind these datasets is that they provide objective image descriptions:

> "By asking people to describe the people, objects, scenes and activities that are shown in a picture without giving them any further information about the context in which the picture was taken, we were able to obtain conceptual descriptions that focus only on the information that can be obtained from the image alone." (Hodosh et al., 2013, p. 859)

Human: Three policemen are standing around someone in a gray sweatshirt with stripes.

Model: A group of people are walking down the street.

Figure 1: Flickr30K image (4944749423) with a human- and a machine-generated description.

The **assumption of neutrality** is a useful simplification: if it is more or less correct that similar images will have similar descriptions (that are not influenced by any external factors), then we can try to learn a mapping between images and descriptions. This is what Vinyals et al. (2015) do. They use a Long Short-Term Memory model to generate sequences of words, given the visual context.[1] Their model is able to produce reasonably good image descriptions without using any higher-order reasoning. Figure 1 provides an example.[2] The machine-generated descriptions are typically shorter and more general than human descriptions. For example, the model talks about 'a group of people', rather than about *a group of policemen* and *a civilian*. Compared to humans, there is less variation in the kind of labels that the model uses to refer to people (section 2.5 of this

[1] The visual context was provided by a convolutional neural network model (Ioffe and Szegedy, 2015), trained for the 2014 ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015).

[2] More examples at https://github.com/evanmiltenburg/NIC-data.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 1–10, Valencia, Spain, April 3-7 2017.

paper). And for a good reason: human-level specificity requires a deeper understanding of context.

This proposal challenges the neutrality assumption, and aims to characterize the subjective nature of image descriptions. Such a characterization is necessary to get an overview of the challenges that lie ahead. My main thesis is that *image description is not a simple mapping from visual features to strings of words. Rather, it is a process involving reasoning, perspective and world knowledge*. This thesis is supported by empirical evidence from image description corpora, showing how the descriptions reflect the crowd-workers' *interpretation* of the images. I will investigate what are the limits of current image description systems, and what is needed in order to get human-like performance, using the model in Figure 2.

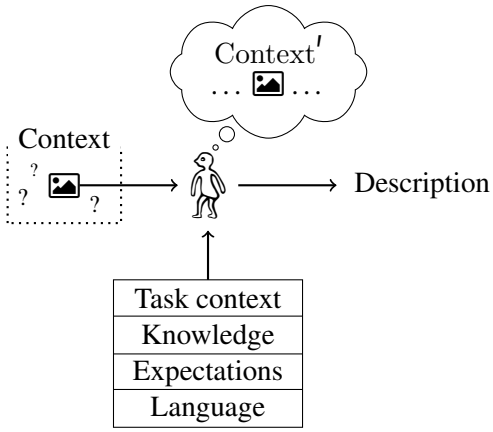


Figure 2: Conceptual model of description generation. Note that the original context is likely to be different from the context inferred by the subject.

In this conceptual model (corresponding to the data collection procedure for Flickr30K and MS COCO), an image is taken out of its original context and presented to a human annotator. Because the original context is lost, the annotator now has to *re-interpret* the image within the context of the task. This new understanding is based on their world knowledge and prior expectations. Next, the annotator has to verbalize his or her understanding of the image in one sentence. This means they have to make choices about (1) which aspects of the picture to focus on, and (2) the way in which those aspects should be described. The first is limited by relevance, whereas the second is limited by the linguistic means afforded by the annotator's native language.

The main goal of this proposal is to better understand the process of describing an image, from the speaker's point of view. In other words: how does someone 'come up' with a description for a given image? How do they determine which features to include? I will use a three-pronged approach to answer these questions:

Corpus analysis Looking for patterns in large volumes of uncontrolled image description data. Our main goal here is to characterize and quantify image description behavior. I will discuss my approach in §2.

Experiment Studying phenomena discovered through corpus analysis in a controlled setting. Our main goal here is to understand what factors drive these phenomena. Corpus analysis and experimentation are two sides of the same coin: we can use a corpus to generate hypotheses about how people describe images, and use experiments to test those hypotheses. See section 3 for more.

Modeling Studying the capacity of automatic image description systems to capture pragmatics. Our main goal here is to characterize the gap between human and machine performance; what makes image description difficult, and how could we face those challenges? Discussed in §4.

The result of this approach is an interdisciplinary picture of the image description process, combining insights from linguistics, natural language processing, and social science.[3] Because of the social relevance of image description systems (one of the main motivations to build these systems is to make images accessible for the visually impaired), I will also discuss the ethical implications of this research (§5). Finally, section 6 takes the first step in generalizing our model to other modalities. Specifically, I discuss similarities and differences between visual and auditory stimuli.

2 Corpus analysis

I will analyze several different image description incorporate in order to characterize the way that people talk about images. The study is not limited to English data sets, but also extends to other languages, allowing us to explore the differences between speakers of different languages. This section provides an overview of the work I have already done (van Miltenburg, 2016; van Miltenburg et al., 2016a, in §2.1–2.4), as well as some work in progress (§2.5,2.6).

[3] Fully in the spirit of Krahmer (2010), who argues that computational linguists can learn a great deal from psychologists (and vice versa).

2.1 What to include in a description

Table 1 shows the ways in which a phrase or expression can be related to an image.[4] It can either refer to something inside or external to the image, and annotators can choose whether or not to use it in their description. This choice is called **content determination/selection** in Natural Language Generation (Reiter and Dale, 1997; Reiter and Dale, 2000). I propose to use this table to systematically study how humans perform this task.

	In the image	**Image-external**
In the description	A	B
Not in the description	C	D

Table 1: Ways in which an expression or a phrase can be related to an image.

The label *policemen* in Figure 1 is an example of **situation A**: there are policemen in the image, and the annotator decided to include that expression in their description. Two other annotators in the Flickr30K dataset also speculated about *an arrest* taking place:

(1) Other descriptions of the image in Figure 1.

a. Three officers arresting someone on the corner of a street.

b. Police officers are arresting a woman.

These are examples of **situation B**, because we cannot conclude this from the image alone. Maybe the person in the gray sweatshirt had just fallen and the officers are helping them stand up. In (van Miltenburg, 2016), I call cases like this **unwarranted inferences**, and provide a list of different kinds of these inferences in the Flickr30K dataset. For example, RELATION-inferences where young children are assumed to be siblings or friends, or women with children are assumed to be mothers.

Traffic light is a nice example of **situation C**: there is a big metal pole right in front of the policemen, but the annotator in Figure 1 made no reference to it. They were also careful not to say that the policemen are *arresting someone*, even though two other annotators did make that inference. This is an example of **situation D**. Note that we are only able to identify this situation because the other annotators *did* speculate about the situation in the image.

[4]For a detailed taxonomy of the inverse relation —ways in which an image can relate to a text, see (Marsh and Domas White, 2003).

2.2 Marking

Following Jakobson (1972) and others in linguistics, we will use the term **marking** to denote the act of signaling an entity or attribute. The difference between Situation A and C is one of *markedness*. We can ask ourselves *why* annotators decide to mark some entities or attributes, but not others. The most basic and naive explanation is that this is because *those are the most important*. But this only gets us part of the way. There is a large amount of variation in the entities and attributes marked by the different annotators in the Flickr30K and MS COCO corpora. An additional explanation is grounded in the work by Beukeboom (2014), who argues that the kind of language people use reflects the way they view the world (he calls this **linguistic bias**); since the annotators' perspectives differ, so do the descriptions.

2.3 What gets marked?

People typically mark entities, properties, or events that are unexpected or go against some social norm (Beukeboom, 2014). Negations are the clearest example of this. Example (2) shows two descriptions from the Flickr30K corpus, where annotators explicitly marked what the subjects in the images *weren't* doing, so as to emphasize that this behavior is unusual.

(2) Examples from van Miltenburg et al. (2016a)

a. Man **not wearing a shirt** playing tennis.
↖ You are supposed to wear a shirt.

b. A boy is eating pie **without utensils**.
↖ You are expected to eat with utensils.

At the same time, there are also structural differences between *a priori* comparable entities or groups of entities in the way they are marked. I annotated all pictures of babies in the Flickr30K dataset, and found that 22% of all black babies are marked as 'black' or 'African-American', and 14% of all asian babies were marked as 'asian' or 'oriental', while less than 1% of all white babies are marked as such (van Miltenburg, 2016). For the group of Flickr30K annotators, it seems that 'white' is the expected default and thus less marked than the others.[5]

2.4 Negations, norms and expectations

As mentioned above, the Flickr30K data contains several descriptions containing negations. The

[5]This is related to reporting bias, see (Misra et al., 2016).

examples in (2) are surprising: somehow crowd workers decided that the best way to describe the relevant images is to say what is missing from them. This behavior is a result of our everyday experience, which (along with social norms) gives rise to expectations about how people are supposed to behave. Negations provide a linguistic means to signal mismatches between our expectations and what is actually happening (Beukeboom et al., 2010). Now consider the items in (3):

(3) a. not wearing a shirt (negation)
b. wearing a blue shirt (specification)
c. wearing a shirt (unmarked)

Negations like (3a) not only signal deviations from the norm, they also (indirectly) tell us *what the norm is*. The same can be said for modifiers further specifying a noun phrase, e.g. (3b); if there are examples of further specification of a noun phrase, but no examples of the 'plain' noun phrase, then we know that the noun phrase corresponds to the norm. Examples like (3c) are typically only used to signal a contrast (in this case: with others not wearing a shirt). I will try to use observations like these to find out what are the implicit norms in image description datasets.

van Miltenburg et al. (2016a) provide a categorization of the different uses of negation, in order to gauge the kind of background knowledge that is required to produce descriptions containing negations. These range from signaling that someone is not wearing a shirt or not using utensils (2) to image-specific cases like (4):

(4) Several people sitting in front of a building taking pictures of a landmark **not** seen.

Here, a crowdworker concluded that the people in the image must be taking pictures *of a landmark*, without having seen the actual landmark. Negations in the Flickr30K data signal cases where world knowledge and commonsense reasoning is required for generating descriptions. This makes descriptions with negations a suitable paradigm to evaluate the extent to which automatic image description systems are able to generate humanlike output. I will check whether state-of-the-art image description systems are able to produce negations and, if so, what kind of negations they are able to produce (see also §4). My expectation is that the production of negations will be limited to common cases, where entire phrases containing negations can be reproduced from the training data. Going beyond those cases requires higher-level reasoning, which current models are not designed to perform.

2.5 Labeling

What kind of labels should be used to refer to people? Figure 3 (next page) shows that there is a large variety of labels in the Flickr30K dataset, that belong to different semantic categories.

Occupation	police officer, businessman, shepherd.
Relation	grandma, boyfriend, colleague, neighbor.
Activity	speaker, activist, presenter. Subcategories:
Sports	snowboarder, athlete, football player
Music	trumpet player, saxophonist, pianist
Age	toddler, boy, girl, adolescent, adult.
Gender	male, female, boy, girl, man, woman.
Appearance	redhead, blonde.
Religion	hindu, muslim, jew.
Other	vagabond, nerd, idiot.

Figure 3: Kinds of person-labels in the Flickr30K dataset, with examples. Subcategories are dominant, coherent subsets of the data.

When annotators decide on a label to use, they can roughly base their judgment on two factors: **appearance** and **situation** of the person to be labeled. Table 2 provides a categorization of person labels in terms of these factors. In the data collection process, I noticed that it was quite easy to find examples of mostly appearance-based or mostly situation-based labels, but difficult to find good examples of labels that seem to depend equally on both appearance and situation. *Civilian* is a good example, because felicitous use of this label requires the relevant person to be around e.g. military personnel (the situation) while not wearing a uniform themselves (appearance).

We can also think of the labels in Table 2 as being on a continuous scale showing the reliance on either of these two factors, as shown in Figure 4. To be clear: I do not want to claim that the use of 'civilian' is somehow *less situation-based* than the use of 'neighbor'. Rather, it balances between two forces that drive the labeling process.

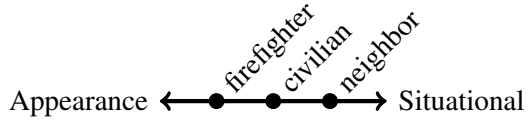


Figure 4: Continuous scale from Appearance-based to Contextually determined labels.

I will further formalize the taxonomy from Figure 3, and extend it to include adjectives and other

Appearance	Situation	Example
Yes	No	Police officer, businessman, firefighter
Yes	Yes	Civilian
No	Yes	Bystander, neighbor, passerby, orphan
No	No	—

Table 2: A categorization of labels based on whether the label is applied on the basis of someone's appearance or the situation they are in.

modifiers, as well as mark each category for its reliance on appearance and situation. I will then study differences in the use of these labels between human annotators and automatic image description systems. We can also use this data as a guide for image description models to produce or not to produce particular kinds of labels. At the same time, this data is useful for natural language understanding as well: with a resource telling us what alternatives a speaker may have in referring to a particular entity, we can reason over *why* the speaker said X while they could have also said Y or Z (Grice, 1975; Geurts, 2010).

2.6 Cross-linguistic analysis

There is a growing interest in collecting image descriptions in different languages, so as to be able to generate descriptions in languages other than English (e.g. Chinese (Li et al., 2016), German (Elliott et al., 2016), Turkish (Unal et al., 2016)). This enables us to study how speakers of different languages describe the same images. Some examples of differences between languages can already be found in the literature. For example: Li et al. (2016) provide the example of an image with a woman taking a picture. In the English descriptions, the woman is referred to as *an Asian woman*, whereas in the Chinese descriptions she is described as a *middle-aged woman* (presumably because *Asian* isn't a distinctive feature in China). Later, the authors note about the English descriptions translated to Chinese that they "do not necessarily reflect how a Chinese describes the same image." This is in line with our model (in Figure 2), which shows the influence of knowledge, expectations, and language on the image description process.

I will study the influence of language by collecting Dutch image description data, and comparing this data with English and German descriptions, so as to see whether the Dutch crowd workers display any behavior that is different from the German and English workers. For example, whether Dutch annotators use different kinds of labels than German or English annotators. The reason for collecting Dutch descriptions is that our project is based in the Netherlands, and if we discover any interesting phenomena, we will be able to carry out additional lab experiments with Dutch participants to further explore those phenomena.

3 Experiment

What makes the crowd describe images the way they do in the MS COCO and the Flickr30K data? I will investigate the degree to which the format of the crowdsourcing task affects the descriptions, and how we can get people to provide different kinds of descriptions. Experiments are essential to test hypotheses that arise from the analysis of image description corpora. I will discuss two experiments below (sections 3.2 and 3.3), but first let us look at the format of image description tasks.

3.1 Canonical format

I will refer to the Flickr30K and MS COCO annotation tasks as the *canonical format*. In this setup, a task consists of a set of general instructions and examples of 'good' and 'bad' descriptions, followed by a set of five images with a prompt to describe each image in one complete, but simple sentence. Crucially, the participants are not told why they are providing the descriptions, or how the descriptions will be used.

3.2 Speculation

Even though the instructions explicitly tell workers not to speculate, we can find many cases of unwarranted inferences. This seems to go against Grice's (1975) Maxim of Quality ("try to make your contribution one that is true"). Assuming that Workers do try to be helpful, my hypothesis is that this behavior is a direct result of the canonical format: left wondering how their description

will be used, Workers just provide as much information as possible because the *question under discussion* is unclear (Roberts, 1996). I plan to test this hypothesis by changing the prompt (specifying how the descriptions will be used) and collecting new descriptions for a subset of the images in the Flickr30K dataset. I expect that the new prompt will make the elicited descriptions more concise and uniform, because participants will focus more on the central aspects of the images that are relevant to the proposed application.

3.3 Entrainment and differentiation

Entrainment and differentiation are well-known effects where speakers either keep re-using the same phrase to refer to the same or similar entities, or change their phrasing to contrast new entities with others (van der Wege, 2009). These within-subject effects have been mostly been studied in the lab with small amounts of abstract examples, and I will use crowdsourcing to extend this research to photographs on a large scale.

To find out whether there are such within-subject effects in the MS COCO and Flickr30K data, it is necessary to know who provided which description, and in what order the images were presented. Because the raw crowdsourcing data with Worker IDs has not been released for the Flickr30K and MS COCO data, we do not know the extent of these effects in image description data. I have contacted the authors to obtain the raw data, and also plan to set up a controlled study to measure entrainment and differentiation effects.

In this study, I will present sets of images in different orders, and collect a large amount of descriptions for each ordering. After collecting this data, I will analyze the data for entrainment or differentiation patterns. This work can also be seen as a more general test of the assumption that the image descriptions in MS COCO and Flickr30K are *independent* from each other. If it turns out that the other images in the task influence the way an image is described, then this effect needs to be taken into account.[6] At the same time, entrainment and differentiation effects are very informative about how people deal with similarity and differences between images, and we should try to see how these effects can be leveraged to create better performing image description systems.

[6]To some extent, this is already controlled for in the current datasets, as Mechanical Turk and Crowdflower randomize crowdsourcing tasks. But this only means that the five descriptions per image are each primed in a different way.

3.4 Related work: Stylistic variation

There is already some prior work showing that the way that crowd workers are prompted for a description can have a strong influence on form of the descriptions. Baltaretu and Castro Ferreira (2016) present results from a study manipulating a crowdsourcing task to get different kinds of referential expressions for the same entity. They experimented with different task prompts within the ReferIt-game (Kazemzadeh et al., 2014). In this annotation game, participants are asked to provide referring expressions for specified entities. They score points if other participants can successfully identify the entity from the referring expressions. Baltaretu and Castro Ferreira (2016) modified the original prompt by asking participants to play fast (FA), be creative (CR), be clear and thorough (CT), or just to provide descriptions without any additional goal (NO). These different prompts had an effect on the length of the expressions (with longer expressions in the CR and CT conditions), and on the amount of adjectives used (with more adjectives in the CR-condition than in the FA-condition). Table 3 shows an example description for each category.

FA	Jumping monkey.
CR	A primate showing off his business end.
CT	Small monkey with a very long tail.
NO	A monkey on a person's head.

Table 3: Example from Baltaretu and Castro Ferreira (2016), showing the difference between the different prompts: Fast, Creative, clear and thorough, and no specific emphasis.

An important observation is that human language is capable of enormous variation. The richness of language poses many challenges to developers of image description systems. For example: when do you use what kind of description?

4 Modeling

Models are essential to our understanding of the world. By building a system that is able to describe an image exactly as a human would do, we can demonstrate that we understand the entire image description process. But right now, we are still far from reaching that goal. In this project, I will try to lay out a road map for the future, by looking

at the discrepancies between human performance and the performance of state-of-the-art models. I plan to carry out three kinds of studies:

Evaluation and error analysis Evaluation of image description systems is typically done by running a metric comparing the generated output with a set of reference descriptions produced by human annotators (see (Kilickaya et al., 2017) for an overview). The problem with these measures is that they are very coarse-grained. I am currently working on a manual error analysis, checking whether automatically generated descriptions are fully congruent with the relevant image, or whether there are any mistakes. Annotating all the mistakes allows us to classify and then quantify which mistakes were made how many times. The error categories show us where there is still room for improvement.

Producing particular phenomena Having made several observations in image description corpora (§2), the question is whether image description systems are able to reproduce those phenomena. For example: can image description systems produce negations? (§2.4) This question calls to mind Chomsky's Competence-Performance distinction (Chomsky, 1965). When image description systems are evaluated on a particular test set, they produce one description for each of those images. This gives us a surface-level idea of their capabilities. But suppose that a system never produces a negation for any image we feed it. That does not mean that the system is incapable of producing negations. Or, putting it in cognitive terms, that it does not *know* how to use negations. It only means that negations are unlikely to be produced by the system. And so we need to dig deeper in order to find out whether the system has gained the relevant knowledge from the training data.

Generating Dutch descriptions Due to the size of the Dutch crowd, I will only be able to collect a relatively small set of Dutch image descriptions. We plan to train a machine translation system that converts English image descriptions to Dutch, so as to extend the Dutch description data. We can then train an image description model for Dutch using this extended dataset. This way we can test whether machine translation is a good strategy to develop image description systems for lesser-resourced languages. I am not the first to propose a translation-based strategy to train an image description system. Li et al. (2016) show that it's possible to train a Chinese system based on translations of the English descriptions from Flickr8K (Hodosh et al., 2013). My contribution will be to provide a qualitative analysis of the system output: does the model make different kinds of mistakes (based on the translation)? Do the descriptions sound natural?

5 Bias and ethics

As recently noted by Hovy and Spruit (2016), there has been "little discourse in the [NLP] community" about ethics, and the social impact of natural language processing. Their paper opens up the discussion, and provides some useful terminology, which can be summarized as follows:

1. Any dataset is **demographically biased**, which may lead to the **exclusion** or **misrepresentation** of social or ethnic groups.
2. Modeling data has the side-effect of **overgeneralization**.
3. "**Topic overexposure** creates biases"; useful heuristics may be disproportionately linked to particular social or ethnic groups.
4. NLP tools could be misused, or (unintentionally) further marginalize particular social or ethnic groups. These are **dual use** problems.

Some of these ideas are also discussed by Gillespie (2014), mostly in the context of information retrieval. He lists six dimensions to critically examine an algorithm, of which we will focus on the **patterns of inclusion**: how is the training data prepared, and what does it contain? We can separate two concerns for the image data under discussion: image selection and annotator selection.

Image selection Both datasets are based on images from Flickr. Gillespie (2014, p. 185) notes that this data may already be biased through users' interaction with the community (who value particular kinds of images) and Flickr's internal search algorithm (which also values particular images and tags). Moreover, images on Flickr also typically depict Western scenes (Miyazaki and Shimizu, 2016).

The Flickr30K images were sourced from six different groups (sub-communities set up around a particular kind of images, e.g. *strangers!* or *dogs in action*) on Flickr, and were manually selected "to depict a variety of scenes and situations" (Hodosh et al., 2013; Young et al., 2014). By focusing

on a small set of groups, one runs the risk of ending up in a 'photo bubble' where the kind of pictures in your dataset is determined by the interests of a small group of people.
The MS COCO images were collected by first compiling a list of object categories, and then looking for images containing those objects on Flickr (Lin et al., 2014). This object-driven approach means that image-sampling takes place at the community level, rather than the sub-community level. A downside of this approach is that it is language-based. Pictures taken by users who don't tag their images or who tag their images in a different language are not considered.

Annotator selection The descriptions of the images for both the Flickr30K and the MS COCO datasets were collected through Amazon's Mechanical Turk. For the former, only workers from the USA who passed a spelling and grammar test were allowed to provide descriptions. No other details about the demographics of the workers were collected. For the latter, Chen et al. (2015) note that their annotation task is strongly inspired by the annotation process for Flickr30K. Again, no details about the demographics of the workers were collected. This makes it very difficult to analyze the data for differences between groups in how they describe an image. We can say that only focusing on workers from the USA means that the descriptions all come from an American point of view. This leads to descriptions like the following, where the Otherness of the images is emphasized (all descriptions taken from the Flickr30K data):[7]

(5) a. This man is looking at shirts in a store where the language is not English .
    b. I see people going into a yellow bus from another country , not United States .
    c. A wild animal not found in America jumping through a field .

To get a sense of the population of Mechanical Turk, Huff and Tingley (2015) carried out a survey among United States workers asking about political attitudes and demographic factors. While there is a reasonably good overall balance between males (54%) and females (46%), the pool is racially skewed with nearly 75% White workers. Of course there might be selection bias in which workers opt for annotation tasks, but these post-hoc numbers are the best we can get. Now recall the finding that that black babies are more often marked as such (using adjectives like *black, African-American*) than white babies (van Miltenburg, 2016). This is consistent with the idea that people typically mark others who are diferent from themselves (mentioned in (Beukeboom, 2014)). Given this social dynamic, it seems clear that annotators should be selected with care. At the very least, it's worth recording more details about the crowd-workers so that we can study the effects of demographic characteristics on image descriptions.

Both image selection and annotator selection give rise to dual use issues. I will focus on the latter, because it hinges on a recurring theme in this proposal: subjectivity in language. If we better understand the processes that give rise to subjective descriptions, then we can also try to mitigate the effects of annotator bias. Through the proposed studies in the previous sections, I aim to raise awareness of the biases in image description data, and to produce a set of tools and resources that will spur improvement in this area. For example, the ability to detect whether or not a description is speculative might help to make systems deliver more factual descriptions.

6 Discussion: Other modalities

We can generalize the observations made about the image description process to other modalities. Distributional approaches to ground language in perceptual data have not only been proposed for images, but also for sounds (Lopopolo and van Miltenburg, 2015; Kiela and Clark, 2015) and even smells (Kiela et al., 2015). We also need to keep these other modalities in mind when we are working on image description, because comparing results for different modalities teaches us what is modality-specific and what is more generally true about the relation between language and perception. As a basis for future work, van Miltenburg et al. (2016b) carried out a crowdsourcing experiment to collect 'keywords' for 2,133 sounds from the Freesound database (Font et al., 2013). For the sounds that were harder to recognize, many participants resorted to speculate about the possible sources of the sound. Really understanding what a sound is about requires annotators to recontextualize the sound and think about likely events

[7] This also works the other way round. When Miyazaki and Shimizu (2016) asked Japanese workers to describe images from Flickr30K, "words such as 'foreign' and 'oversea' [initially were] everywhere in the descriptions" (p. 1783).

that may have caused it. The difference between sounds and images is that sounds are dynamic (and thus contain more information about actions than about entities) while images are static (and thus contain more information about entities).

7 Conclusion

In this paper I have proposed to study the image description process in terms of the model in Figure 2, using three different approaches: corpus analysis, lab experiments, and using image description models. This work will hopefully lead to a more complete characterization of the knowledge that human annotators bring to bear on image description tasks. This characterization will provide a road map to make automatic image description systems display more human-like behavior.

Acknowledgments

I would like to thank the members of the CLTL lab, and two anonymous reviewers for their valuable feedback. Many thanks to all my collaborators. This project is supervised by Piek Vossen and Desmond Elliott. All faults are my own. This research was supported by the Netherlands Organisation for Scientific Research (NWO) via the Spinoza grant, awarded to Piek Vossen.

References

Adriana Baltaretu and Thiago Castro Ferreira. 2016. Task demands and individual variation in referring expressions. In *Proceedings of the 9th International Natural Language Generation conference*, pages 89–93, Edinburgh, UK, September 5-8. Association for Computational Linguistics.

Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55:409–442.

Camiel J. Beukeboom, Catrin Finkenauer, and Daniël H. J. Wigboldus. 2010. The negation bias: when negations signal stereotypic expectancies. *Journal of personality and social psychology*, 99(6):978.

Camiel J. Beukeboom. 2014. Mechanisms of linguistic bias: How words reflect and maintain stereotypic expectancies. In J. Laszlo, J. Forgas, and O. Vincze, editors, *Social cognition and communication*, volume 31, pages 313–330. Psychology Press. Author's pdf: `http://dare.ubvu.vu.nl/handle/1871/47698`.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5^{th} Workshop on Vision and Langauge at ACL '16*.

Frederic Font, Gerard Roma, and Xavier Serra. 2013. Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 411–412. ACM.

Bart Geurts. 2010. *Quantity implicatures*. Cambridge University Press.

Tarleton Gillespie, 2014. *Media technologies: Essays on communication, materiality, and society*, chapter The Relevance of Algorithms, pages 167–193. MIT Press.

Herbert Paul Grice. 1975. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and semantics 3: Speech acts*, pages 44–58. Academic Press, New York.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Dirk Hovy and Shannon L. Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, Berlin, Germany, August. Association for Computational Linguistics.

Connor Huff and Dustin Tingley. 2015. "who are these people?" evaluating the demographic characteristics and political preferences of mturk survey respondents. *Research & Politics*, 2(3):2053168015604648.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Roman Jakobson. 1972. Verbal communication. *Scientific American*, 227:72–80.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar, October. Association for Computational Linguistics.

Douwe Kiela and Stephen Clark. 2015. Multi- and cross-modal semantics beyond vision: Grounding in auditory perception. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2461–2470, Lisbon, Portugal, September. Association for Computational Linguistics.

Douwe Kiela, Luana Bulat, and Stephen Clark. 2015. Grounding semantics in olfactory perception. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 231–236, Beijing, China, July. Association for Computational Linguistics.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *To appear in Proceedings of EACL 2017*. Available as arXiv preprint arXiv:1612.07600.

Emiel Krahmer. 2010. What computational linguists can learn from psychologists (and vice versa). *Computational linguistics*, 36(2):285–294.

Xirong Li, Weiyu Lan, Jianfeng Dong, and Hailong Liu. 2016. Adding chinese captions to images. In *Proceedings of the 2016 ACM International Conference on Multimedia Retrieval*, pages 271–275. ACM.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer.

Alessandro Lopopolo and Emiel van Miltenburg. 2015. Sound-based distributional models. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 70–75, London, UK, April. Association for Computational Linguistics.

Emily E. Marsh and Marilyn Domas White. 2003. A taxonomy of relationships between images and text. *Journal of Documentation*, 59(6):647–672.

Ishan Misra, C. Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. 2016. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2939, June.

Takashi Miyazaki and Nobuyuki Shimizu. 2016. Cross-lingual image caption generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1780–1790, Berlin, Germany, August. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(01):57–87.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Studies in Natural Language Processing. Cambridge University Press.

Craige Roberts. 1996. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Working Papers in Linguistics-Ohio State University Department of Linguistics*, pages 91–136.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Mesut Erhan Unal, Begum Citamak, Semih Yagcioglu, Aykut Erdem, Erkut Erdem, Nazli Ikizler Cinbis, and Ruket Cakici. 2016. Tasviret: Görüntülerden otomatik türkçe açıklama oluşturma İçin bir denektaçı veri kümesi (tasviret: A benchmark dataset for automatic turkish description generation from images). In *IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU 2016)*.

Mija M. van der Wege. 2009. Lexical entrainment and lexical differentiation in reference phrase choice. *Journal of Memory and Language*, 60(4):448–463.

Emiel van Miltenburg, Roser Morante, and Desmond Elliott. 2016a. Pragmatic factors in image description: the case of negations. In *Proceedings of the 5^{th} Workshop on Vision and Language at ACL '16*.

Emiel van Miltenburg, Benjamin Timmermans, and Lora Aroyo. 2016b. The vu sound corpus: Adding more fine-grained annotations to the freesound database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association (ELRA).

Emiel van Miltenburg. 2016. Stereotyping and bias in the flickr30k dataset. In Jens Edlund, Dirk Heylen, and Patrizia Paggio, editors, *Proceedings of Multimodal Corpora: Computer vision and language processing (MMC 2016)*, pages 1–4.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Detecting spelling variants in non-standard texts

Fabian Barteld

Institut für Germanistik / Language Technology Group, Department of Informatics
Universität Hamburg
firstname.lastname@uni-hamburg.de

Abstract

Spelling variation in non-standard language, e.g. computer-mediated communication and historical texts, is usually treated as a deviation from a standard spelling, e.g. *2mr* as a non-standard spelling for *tomorrow*. Consequently, in normalization – the standard approach of dealing with spelling variation – so-called non-standard words are mapped to their corresponding standard words. However, there is not always a corresponding standard word. This can be the case for single types (like emoticons in computer-mediated communication) or a complete language, e.g. texts from historical languages that did not develop to a standard variety. The approach presented in this thesis proposal deals with spelling variation in absence of reference to a standard. The task is to detect pairs of types that are variants of the same morphological word. An approach for spelling-variant detection is presented, where pairs of potential spelling variants are generated with Levenshtein distance and subsequently filtered by supervised machine learning. The approach is evaluated on historical Low German texts. Finally, further perspectives are discussed.

1 Introduction

Spelling variation is a well-known feature of non-standard language, e.g. computer-mediated communication (CMC) and historical texts (Baron et al., 2009; Eisenstein, 2013). One problem is that this variation decreases the utility of unannotated corpora, e.g., by reducing the recall for queries and distorting keyword frequencies (Baron et al., 2009). Furthermore, the variation makes it harder to annotate this data automatically since the number of out-of-vocabulary (OOV) words is higher when compared to the same amount of standardized data. In addition, during training time, instances of the same morphological word appear as different types thereby distributing the information about one morphological word over these types.

The predominant way to deal with spelling variation is normalization, i.e. non-standard words are mapped to a corresponding standard word, or a canonical form. In this thesis proposal, we pursue an alternative approach: the task of spelling-variant detection, i.e. instead of mapping non-standard or historical words to a standard form as in normalization, the aim is to detect spelling variants in a set of types without reference to a canonical form. Therefore, this task can be applied in cases where no canonical form exists. The detected spelling variants can then be used to mitigate the problems caused by spelling variation that were described above. An approach for detecting spelling variants is presented and evaluated on Middle Low German (GML), a group of German dialects from between 1200 and 1650. These dialects developed into Low German, a dialect group of German that has not undergone standardization. Therefore, there exists no contemporary variant of Low German with standardized orthography that could be used as target language.

After having discussed related work in Section 2, we elaborate on the task of spelling-variant detection (Section 3) and introduce an approach using binary classification (Section 4). We present first experiments on generating candidate pairs (Section 4.1) and filtering these pairs using a supervised machine learning approach (Section 4.2). Finally, we conclude with an outlook on further planned research in the framework of this thesis.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 11–22, Valencia, Spain, April 3-7 2017.

2 Related work

In normalization – also called standardization (Ljubešić et al., 2014) or canonicalization (Jurish, 2010a) – spelling variation is seen as a deviation from a given standard. In the case of CMC this is the standardized language as used in newspapers and regarding historical texts this is the corresponding contemporary standard language. This kind of normalization has been criticized as a "lossy translation" for CMC data (Gimpel et al., 2011) and it has been shown that it is not capable to deal with all peculiarities that appear in non-standard texts.

An example for this is that normalization cannot deal with differences in the usage of a word between the standard and the non-standard domain that result in different labels – a tagger trained on the standard domain will apply the wrong tag. Consequently – as Yang and Eisenstein (2016) show – the accuracy of tagging historical texts benefits from combining domain adaptation and normalization.

Such differences in the usage of a word are also visible in semantic changes. Bollmann et al. (2012) therefore distinguish between normalization and modernization: While in the first a historical word is mapped to its modern cognate, in the second the historical word is more loosely translated. They give the example of the Early New High German (1350-1650) word *vrlaub* 'permission' which would be normalized to its Modern German cognate *Urlaub* 'vacation' but modernized to the Modern German word *Erlaubnis* 'permission'. Another, more subtle example for a lossy normalization, would be *Kopf* and *Haupt*. Both denote 'head' in Modern German, but *Haupt* is only used in exalted language. However, in the Middle High German period (1050-1350), *houbet* the cognate of *Haupt* was the most commonly used word while the cognate of *Kopf* was mainly used in descriptions of battles (Fritz, 2006). Such differences will not be relevant for tasks like POS tagging. Still, this example shows the lossy nature of normalization to a modern standard.

Another limitation of normalization is that it can only deal with items that have a corresponding standard word. This can be solved by creating an artificial standard for the specific phenomenon and normalizing towards this standard. Dipper (2011) has shown that using an artificial standardized version of Middle High German leads to better results training and applying POS and morphological taggers to these texts. In CMC, items like emoticons have no corresponding standard form and require a special treatment when normalizing these texts. E.g., for the shared task of normalizing Twitter data (Baldwin et al., 2015) only all-alphanumeric tokens are normalized. This excludes tokens like *=)*, *:)* and *:-)* from the normalization. One way to deal with variation in emoticons is again the normalization to an artificial standard, cf. the manual mapping of different emoticons to synsets used by Hogenboom et al. (2015). Hence, normalizing to an artificial standard solves the problems of normalization. However, this introduces the need to develop the standard first.

These problems do not appear when detecting spelling variants without the reference to a standard. The detection of spelling variants has been applied in a minor strand of work on automatically annotating non-standard texts. For POS tagging historical texts, the knowledge about spelling variants has been used to substitute OOV words with possible spelling variants – this improved the accuracy of POS taggers trained on the non-standard data (Logačev et al., 2014; Barteld et al., 2015). A similar approach has been used by Gadde et al. (2011) for SMS text. In all of these approaches the knowledge about spelling variants is used independently from the tagger, either in a pre-processing step to reduce the amount of spelling variation before training and/or tagging or as a post-processing step to correct tagging errors.

Alternatively, specialized tools can be developed that directly use the knowledge about spelling variation. This approach has been followed by Kestemont et al. (2010) and Barteld et al. (2016) for lemmatization of historical texts.

With the exception of Acharyya et al. (2009), who present a clustering approach based on context similarity that also takes surface similarity into account, spelling-variant detection has not been addressed independently of a specific task like POS tagging. However, Acharyya et al. (2009) only present an anecdotal evaluation of resulting clusters.

Similar to the task of detecting pairs of spelling variants is the detection of pairs of standard and non-standard words appearing in noisy text and the detection of cognate pairs. Usually, surface similarity, contextual similarity or a combination of both is used for these tasks. Gouws et al. (2011)

use the bigrams that appear before and after a word to compute distributional similarity and string kernels to measure surface similarity to create a dictionary of non-standard and corresponding standard types.

3 Defining spelling variation and spelling-variant detection

In order to define spelling variation, we distinguish between *type* and *morphological word*. Morphological words are the inflected word forms of a language. This distinction is similar to the distinction between *morphological word* and *lexeme*: A lexeme like the English verb *(to) be* appears as different morphological words in texts, e.g. *am* and *was*. In lemmatization, the task is to assign the same lemma to the different morphological words of the same lexeme, thereby abstracting over inflectional differences. Similarly, the aim of spelling-variant detection is to detect the types that belong to the same morphologicial word. However, while the morphological words belonging to the same lexeme differ with respect to their morphology, the types that belong to the same morphological word only differ in their spelling.

Consequently, the notion morphological word can be operationalized by combining POS, morphological information, lemma and word-sense disambiguation: Each combination of these attributes is a morphological word. For example {*Personal Pronoun, Nominative Sg., I*} is a morphological word of English. In texts, these abstract word forms are realized by using types. Types are the words t of a language $L \subseteq \Sigma^*$ for some given alphabet Σ. For a standardized language, a unique mapping from a morphological word to a type is expected.[1] For instance, {*Personal Pronoun, Nominative Sg., I*} is usually realized as 'I'. In GML texts, the corresponding morphological word can be realized using the types given in Example 1.

(1) yck, ick, jk, ik, yk, jck, jc, ic

This example shows *i*, *j* and *y* that appear in general mostly interchangable in GML manuscripts and prints. Only bi-graphs like *ei* are an exception to this rule. Niebaum (2000) gives an overview over the graphematic inventory of GML. He lists the grapheme <*i*> and amongst others *j* and *y* as variants. He states that these are often used instead of *i* to disambiguate the spelling next to letters like *m* and *n*. However, the example of 'I' shows that they are used interchangeably in other contexts as well. Such variation can be easily modelled in a rule-based approach. We will compare our approach with a rule-based approach developed for GML.

Variation as in Example 1 is what we define as spelling variation in a broad sense: the realization of a morphological word using different types. There is spelling variation even in standardized texts. One example is spelling errors that lead to a variation. But there is also real spelling variation in standardized language. One example for this rare case where two different types are used for the same morphological word in standardized language is the co-existence of *ß* and *ss* in Modern German which leads to the spelling variants *Fuß* and *Fuss* '(the) foot'. This is, however, negligible for standard texts: only 7% of the morphological words appearing more than once in the Tiger corpus (Brants et al., 2004), a corpus consisting of German newspaper texts, show variance, i.e., are realized by more than one type.[2]

In non-standard texts, there is more variation: In the English Twitter texts used as the training data for the W-NUT 2015 shared task on normalization (Baldwin et al., 2015), 57% of the morphological words show variation.[3] This can be reduced to 16% by lowercasing every type.

Historical texts, for which the amount of data available is often extremely small, also show a lot of variation. In the GML texts from the Reference Corpus Middle Low German (Peters and Nagel, 2014) that are used for the experiments in this paper[4], 58% of the morphological words show variation. However, less of this variation is due to differences in the case as in the Twitter data: If every

[1] Not the other way around, due to ambiguity. The German type *Bank* can either denote the nominative singular of *bench* or *bank*. However, the German morphological word *bench* (nominative singular) is always realized as *Bank*.

[2] As the Tiger corpus is not annotated with word-sense disambiguation, this overestimate the actual amount of spelling variation in the corpus as e.g. the types *Bänke* ('benches') and *Banken* ('banks') are incorrectly treated as the same morphological word {*Noun, Nominative Pl., Bank*}. Tokens tagged with FM, XY, OA, PTKVZ or one of $., $, and $(have been excluded for the calculation.

[3] As the W-NUT 2015 data is neither annotated with POS, morphology nor lemma, we use the normalization annotation: Each normalization is treated as a morphological word. Tokens containing non-alphanumeric characters have been excluded as they were not normalized.

[4] We use the release 2016-08-23 of the corpus (`http://hdl.handle.net/11022/0000-0001-B002-5`).

type is lowercased still 54% of the morphological words show variation.

These numbers quantify the internal – or synchronic, cf. Piotrowski (2012) – variation in the corpus and thereby indicate the amount of the data sparsity problem. Consequently, they give an indication of how hard it is to develop tools for this language variant. This is different than the usually reported number of OOV types with respect to a dictionary of the corresponding standard. The latter only indicates how promising it is to apply tools developed for the standard to the language variant in question.

Normalization does not deal with spelling variation directly but with non-standard words, a term used by Sproat et al. (2001), who introduced the task of normalization. For historical text, the non-standard words are historical words or historical forms of contemporary words – the standard words are contemporary words. This approach only looks at diachronic variation and not at synchronic variation as defined above. Internal variation in the data is only dealt with indirectly by mapping the non-standard types to a corresponding standard type. Hence, it resembles a translation task, a framework in which normalization has been approached (Kobus et al., 2008; Scherrer and Erjavec, 2016). The task of detecting spelling variants shifts the attention towards the internal variation and resembles an information retrieval task where the aim is to detect unordered pairs of types like GML {*jc*, *ik*} which are used to realize the same morphological word. More formally, given a set of types $L \subseteq \Sigma^*$, the aim is to retrieve all pairs of types that are spelling variants, i.e. the set $SV \subseteq \{\{t_1, t_2\} | t_1 \in L, t_2 \in L, t_1 \neq t_2\}$. SV defines the spelling-variant relation.

As has been noted, spelling variation as defined in this paper is a broad cover term for different types of variation that appear between types for which the spelling-variant relation holds. In order to give an overview over the phenomena that have to be covered for spelling-variant detection, the following constructed pair of GML sentences illustrates three different types of spelling variation.

(2) Do he komen was van deme kloster
DO he ghecomen was uan dem kloster
when he come.PPTC was from the cloister
'when he had been coming from the cloister'

{*Do*, *DO*} and {*van*, *uan*} from Example 2 illustrate spelling variation in the narrow sense, i.e. two types for which the same pronunciation is assumed. However, spelling variation in the broad sense also covers {*deme*, *dem*} where the final <e> is assumed to correspond to the pronunciation of a final [ə]. Finally, there is morphological variation as in {*komen*, *ghecomen*} where the types differ by the spell-out of the morphological marker *ghe* in the participle *komen* '(to) come'.

A clear-cut distinction between those three types of variation is not always possible: {*deme*, *dem*} could also be a spelling variant in the narrow sense, as it cannot be decided for a single instance if there actually was a difference in the pronunciation. Furthermore, the difference between {*deme*, *dem*} could also be classified as morphological variation, treating the *e* as the overt dative marker.

As has been pointed out above, we cover errors under the term spelling variation as well. While errors are usually defined as a deviation from a norm (Brill and Moore, 2000), in the case of the lack of a norm, we define them as a type of variant that is unlikely to appear, it may even appear only once. The GML corpus, for instance, contains one instance of *gesprok* as the past participle of *speak*, whereas other instances of the past participle are realized with the suffix *en*. In this example it is likely that this suffix was to be realized as an abbreviation, i.e. a dash over the *k*, and was simply forgotten.

Besides actual spelling errors, for historical texts another source of spelling variation are errors in the transcription (done manually or with optical character recognition). These lead to variants that should be discovered by the algorithm as well.

4 Approaches towards spelling-variant detection

For the experiments presented in this paper, we use five texts from the Reference Corpus Middle Low German (Peters and Nagel, 2014). The corpus is annotated with POS, morphological information, lemma and word-sense disambiguation. In order to exclude temporal and spatial variants, the texts are taken from the same dialect region and roughly from the same time (about 1500 AD). The texts consist of 36,269 tokens. We use two texts that constitute about 80% of the tokens ('Buxteh. Ev.', 19,644 tokens and 'Griselds', 9,057 to-

kens) for training, the other three texts ('Veer Koeplude', 4,691 tokens, 'Agneta Willeken', 2,269 tokens and 'Reval Tot.', 608 tokens) are each split into two halves, using the first halves as development and the second halves as test set. Pairs of types that are instances of the same morphological word, i.e. they have the same POS, morphology and lemma containing word-sense information, are extracted from these texts. Tokens, for which the annotation groups together types that are not spelling variants were removed from the the corpus. This includes text that is struck through in the manuscript. Such tokens were always annotated with the tag OA ('no annotation'). This reduces the number of training tokens to 26,915, the size of the development set to 3,393 tokens and the size of the test set to 3,396 tokens. We report precision, recall and F-score for the set of spelling-variant pairs extracted from the test set that did not appear in the training data. These are 68% of the pairs. This way of splitting the data into training and test set makes the task harder than directly splitting the set of spelling-variant pairs as the amount of pairs containing rare words will be higher in this setting. However, this way of evaluating the task will give a more realistic estimation of the performance for applications like POS tagging for which the main task of spelling-variant detection is exactly to identify variants of OOV words in new texts.

We approach spelling-variant detection in two steps: First, we generate pairs of spelling variants, then we employ a supervised binary classifier to filter out overgenerated pairs. The generation step is employed in order to reduce the number of pairs that have to be classified. Without this step, for a given set of types L each of the $\binom{n}{2}$ pairs $\{t_1, t_2\} \in L \times L$ would have to be classified, which is computationally intractable for large sets. This pair generation step needs to be fast while having a high recall for actual spelling variants.

4.1 Candidate-pair generation

For candidate-pair generation, we rely on surface similarity between the types. From the set L, we generate all pairs of types for which the Levenshtein distance (Levenshtein, 1966) is below a given threshold s. There are several efficient approaches for detecting all types from L for which the distance is below s, some have even been proposed in the context of normalization, e.g. anagram hashing (Reynaert, 2009). We use a Levenshtein automaton (Schulz and Mihov, 2002) to retrieve all the pairs of types from L that have a distance below $s \in \{1, 2, 3, 4\}$. Table 1 shows recall, precision and F-score as well as the average number of candidate pairs per type (arithmetic mean and standard deviation) for the different values of s. The numbers show that most of the spelling variants have a distance smaller than or equal to 3. Going to distance 4 improves recall from 0.97 to 1 but also increases the average size of generated candidate pairs from about 83 to 261 per type. The precision is always very low, even at a Levenshtein distance of 1 where the average number of predicted variants is slightly below 2. This is due to the fact that many types do not have spelling variants.

For cognate recognition as well as mining pairs of standard and non-standard words, variants of a weighted Levenshtein distance have been used in order to increase recall and precision, e.g. by Hauser and Schulz (2007) for detecting historical variants of modern words and Gomes and Lopes (2011) in cognate detection. These approaches usually employ quasimetrics, i.e. the used metrics are not necessarily symmetric as the weights learned for edit operations are learned in one direction – $i \rightarrow y$ may have a different cost than $y \rightarrow i$. This makes sense in the context of normalization and cognate detection as the comparisons made in these cases are directed as well, e.g., types from older stages of a language are compared to the modern variant. However, in the case of spelling-variant detection the weights for both directions should be equal because next to the pair {*ghecomen*, *komen*} from Example 2 the pair {*ghekomen*, *comen*} exists where the difference *c* $\leftrightarrow$ *k* appears in opposite directions.

We use an undirected version of the measure SPSim by Gomes and Lopes (2011) as a baseline for our experiments. This measure employs substitution patterns (SP), i.e. segments of mismatches from the alignment of the types in the candidate pair with their left and right context. Example 3 shows a pair of spelling variants and the corresponding undirected SP with a context of length 2 denoted by the triple (left context, {pair of mismatched characters}, right context). The context is padded with *$* at the beginning and the end of the type.

(3) maria, marien
('ri', {'a', 'en'}, '$$')

Lev	R	P	F1	AVG	SD
1	0.58	0.12	0.20	1.85	2.42
2	0.88	0.02	0.04	15.99	20.06
3	0.97	0.00	0.01	83.39	84.78
4	1.00	0.00	0.00	261.41	202.86

Table 1: Recall and Precision using the Levenshtein distances
Levenshtein distance (Lev), Recall (R), Precision (P), F-score (F1), Number of candidates per type: average (AVG) and standard deviation (SD)

Lev	R	P	F1	AVG	SD
1	0.48	0.18	0.26	1.11	1.51
2	0.65	0.09	0.15	2.93	3.92
3	0.66	0.05	0.10	4.54	6.11
4	0.67	0.05	0.09	5.08	6.58

Table 2: Results using undirected SPSim (0.9)
Levenshtein distance (Lev), Recall (R), Precision (P), F-score (F1), Number of candidates per type: average (AVG) and standard deviation (SD)

The measure is trained on positive examples. When applying SPSim, SPs that appeared in the training data get a cost of 0, otherwise their cost is the edit distance between the mismatched segments. Furthermore, the context is generalized, i.e. when a mismatch segment appears in the training data with at least two different contexts, the mismatch will always get a cost of 0 regardless of the context.

Using this measure, pairs of types where all the changes are known get the maximal similarity of 1. This allows for improving the precision without losing on recall by setting a high threshold on the similarity for cognate detection. The results using the undirected version of SPSim for identifying spelling variants in the GML test set can be seen in Table 2. The threshold of 0.9 produced the best results on the development set.

While there is an improvement in F-score from 0.20 to 0.26, the precision is still very low (0.18). One reason for this is that the training data contains a lot of very generic substitutions that are learned by SPSim. The following SPs are the SPs that are learned from the training data and involve a single 'a': ($\emptyset$, {'a', 'o'}, $\emptyset$), ($\emptyset$, {'a', 'u'}, $\emptyset$), ($\emptyset$, {'a', 'e'}, $\emptyset$) and ($\emptyset$, {'a', 'en'}, $\emptyset$).

With this generic set of SPs, two types like *dach* '(the) roof' and *doch* 'but' differing only in *a* vs. any other vowel except for an *i* will have a similarity of 1. The pattern ($\emptyset$, {'a', 'u'}, $\emptyset$) is learned from the two spelling variants {*sundighe*, *sandige*} and {*ghehat*, *ghehut*}. However, the first pair is likely to be an error in the original manuscript, the second example is an error in the gold annotation, leading to a wrongly learned pattern.

In order to make the classification more robust against such noise in the data, a more complex weighting scheme for SPs than 0 and 1 should be used. We follow the approach that Ciobanu and Dinu (2014) apply to cognate recognition by training a binary classifier on positive and negative examples for spelling variants to filter out overgenerated candidate pairs.

4.2 Filtering overgenerated candidate pairs

For filtering out overgenerated candidate pairs, we apply a binary classifier that is trained on positive and negative examples of pairs of types. We experiment with two different kinds of features: surface features – representing similarities and differences in the strings – and context features.

As surface features we use undirected SPs as defined in the previous section as well as paired character n-grams around mismatches (Ciobanu and Dinu, 2014), and all paired character n-grams (Ciobanu and Dinu, 2015) extracted from the aligned sequences, see Example 4.

(4) maria, marien
2-grams: {$m, $m}, {ma, ma}, ...,
{ia, ie}, {a_, en}, {_$, n$}
2-grams(mis): {ia, ie}, {a_, en}, {_$, n$}

For the n-grams we test all combinations of lengths in $\{1, 2, 3\}$. Similarly, for the SPs we use context sizes of $\{0, 1, 2\}$. Furthermore, we combine the n-grams with SPs.

As context feature, we use the cosine similarity between dense vector representations (vec) obtained using positive pointwise mutual information with singular value decomposition on a larger unannotated set of GML texts (1,730,614 tokens) than the annotated texts used for the experiments. As suggested by Levy et al. (2015), we have tested different hyperparameters for the creation of the dense vectors: dimensions ($\{125, 250, 375, 500\}$), context windows ($\{2, 5\}$) and frequency thresholds ($\{10, 25, 50, 75, 100\}$). We have also combined the context feature with the best performing surface features and the surface features with the best performing context feature.

For classification a Support Vector Machine (SVM) is trained. We use a radial basis function (RBF) kernel and train the model with the Weka (Witten et al., 2011) wrapper for LibSVM (Chang and Lin, 2011) doing a grid-search over the values $\{1, 2, \ldots, 5\}$ and $\{10^{-2}, \ldots, 10^{2}\}$ for the hyper-parameters c and γ on the development set.

The classifier is trained on positive and negative examples. As positive examples, we use all the pairs of spelling variants appearing in the training data (1834 pairs). In order to obtain negative examples, we extract pairs of types with a Levenshtein distance of 1 and of 2 that do not appear with the same annotation using only types that appear at least 10 times in the training data. The frequency threshold is used to reduce the probability that the pair is actually a pair of spelling variants that – due to ambiguity of the types – did not occur as the same morphological word in the training data. The negative pairs are sampled randomly to obtain the same number of negative and positive pairs. In the sampling procedure, we prefer pairs with a lower Levenshtein distance.

We apply the trained classifier to all candidate pairs obtained using the generation process described in the previous section using the Levenshtein distances $1, 2$ and 3. Table 3 shows relevant results.

Overall, all of the features lead to an improvement in F-score over the best F-score obtained using the Levenshtein distance (0.20) and the undirected SPSim (0.26). Combining the different types of surface features did not improve the results.

Differing from the result obtained by Ciobanu and Dinu (2015) for discriminating between cognates and borrowings, using only n-grams around mismatches leads to better overall result than using only n-grams in terms of F-score (0.38 against 0.36), but using all n-grams leads to a slightly better recall (0.43 against 0.42). Both features lead to better results than using SPs, which lead to an F-score of 0.34. However, the differences between these three feature types are small and are not stable across different splits of the dataset.

Using only context features, the results are comparable to the results with surface features, regarding the F-score (0.36). However, this F-score results from a higher recall and a lower precision. A context size of 2, a small frequency threshold (10) and the dimensions 500 and 375 lead to the best results on the data set.

Combining surface and context features results in the best F-score (0.42) using this approach. However, in experiments with vectors obtained from a smaller subcorpus (739,576 tokens), adding the context features led to no improvement over using only surface features.

Regarding the generation method, the best F-scores are obtained using a Levenshtein distance of 1. The increase in recall obtainable by adding further candidate pairs corresponds to a larger drop in precision.

Finally, we compare our approach with a rule-based approach. For this, a set of 26 rules developed by linguists for the purpose of reducing the spelling variation in GML texts is used to detect spelling variants. The rules consist of regular expressions and substitutions. They are applied in a fixed order.[5] Example 5 gives an exemplary rule. Example 6 gives the set of spelling variants for the personal pronoun in first person singular (Engl. 'I') and the remaining variants after applying the rules showing that the number of variants for this morphological word is reduced from 8 to 2 by the rule-based approach.

(5) /ck?(?!h)/ → /k/

(6) {ik, ic, ick, jc, jck, jk, yck, yk} → {ik, jk}

All pairs of types that are mapped to the same form by applying these rules are considered spelling variants. With this approach a slightly better F-score as with SPSim is obtaind (0.29), see Table 4. However, it is outperformed by the machine learning approach. By simply taking the union of the sets of spelling variants obtained using the rules and the best binary classification model, we obtain the best F-score (0.45), see Table 4.

5 Conclusion and future work

In this paper, we presented the task of spelling-variant detection and preliminary results of an approach using supervised machine learning, which was evaluated on Middle Low German texts. The

[5]The script applying these rules to the data has been created by Melissa Farasyn in the project 'Corpus of Historical Low German' (CHLG; `http://www.chlg.ac.uk/index.html`) and contains rules by Melissa Farasyn with additions by Sarah Ihden and Katharina Dreessen both from the project 'Reference Corpus Middle Low German/ Low Rhenish (1200-1650)'.

Lev	Features	R	P	F1	AVG	SD	c	γ
1	n-gram(mis): 1, 2, 3	0.42	0.34	0.38	0.62	0.86	2	10^{-1}
1	n-gram: 1, 2, 3	0.43	0.31	0.36	0.68	0.91	3	10^{-1}
1	SP: 0, 1	0.37	0.31	0.34	0.60	0.84	2	10^{0}
1	vec: 500, 2, 10	0.52	0.28	0.36	0.83	1.08	4	10^{-1}
1	vec: 500, 2, 50, n-gram(mis): 1, 2, 3	0.47	0.37	0.42	0.62	0.87	2	10^{-1}
1	vec: 375, 2, 25, n-gram(mis): 1, 2, 3	0.47	0.38	0.42	0.62	0.86	2	10^{-1}
2	vec: 500, 2, 10, n-gram: 3, n-gram(mis): 1, SP: 2	0.58	0.17	0.26	1.48	1.81	4	10^{-1}

Table 3: Recall and Precision for the binary classification approach
Levenshtein distance (Lev), Recall (R), Precision (P), F-score (F1), Number of candidates per type: average (AVG) and standard deviation (SD), hyperparameters for the SVM (c, γ)

Method	R	P	F1	AVG	SD
Rule	0.19	0.67	0.29	0.20	0.56
SVM	0.47	0.38	0.42	0.62	0.86
SVM+Rule	0.52	0.39	0.45	0.67	0.93

Table 4: Recall and Precision for the rule-based approach and the combination with the best binary classification
Recall (R), Precision (P), F-score (F1), Number of candidates per type: average (AVG) and standard deviation (SD)

Lev	R	P	F1	AVG	SD
1	0.36	0.03	0.05	3.10	6.62
2	0.65	0.00	0.01	48.49	84.93
3	0.83	0.00	0.00	299.57	368.07
4	0.92	0.00	0.00	913.19	785.16

Table 5: Recall and Precision using the Levenshtein distance for English Twitter data
Levenshtein distance (Lev), Recall (R), Precision (P), F-score (F1), Number of candidates per type: average (AVG) and standard deviation (SD)

results obtained are better than using a variant of the trainable edit distance SPSim that was developed for cognate detection. Furthermore, this approach outperformed a rule-based approach with rules developed by linguists for the GML data. Still, the overall F-score obtained is low. In the proposed thesis, we will focus on various ways to improve these results (Section 5.1). In addition, we will extend the scope of the approach (Section 5.2) and use extrinsic evaluations (Section 5.3).

5.1 Improving spelling-variant detection

Improving the precision of the generator seems like a promising way to improve the results, as the drop in precision when going from Levenshtein distance 1 to 2 for generating candidate pairs, led to worse results regarding the F-score.

We will also look into ways to improve the context features, e.g., by using vector representations obtained from the skip-gram (Mikolov et al., 2013) or other models. As for historical texts the amount of texts available is often very limited, we will experiment with ways to improve the obtained vector representations from smaller data sets, e.g., by taking surface similarity into account as Acharyya et al. (2009) do in their clustering approach, by improving the representations of rare words (Sergienya and Schütze, 2015) which are especially important in spelling-variant detection or by using only the most frequent types in the context (Gimpel et al., 2011).

5.2 Extending the scope

For this paper, we limited the data used for training and evaluation to GML texts from only one dialect region and the same time. In future work, we will expand the scope of variant detection. We will add data from other dialect regions and time spans, which will add dialectal and diachronic variation as well – which is common for historical corpora that contain heterogeneous texts.

We will also extend the experiments to other kinds of non-standard data, especially CMC texts. We did first experiments with Twitter data using the data of the W-NUT 2015 normalization shared task (Baldwin et al., 2015) and treated all types that are normalized with the same type as spelling variants. Table 5 shows results for generating candidate pairs using the Levenshtein distance. There is a difference when comparing these results to the results obtained for the GML data (see Table 1): while the number of candidates generated using the Levenshtein distance is higher, at the same time the recall is lower. Therefore, we plan to experiment with other ways of generating candidate pairs in order to reduce the number of pairs that have to be classified, e.g., by using a distributional thesaurus (Riedl and Biemann, 2013). A surface

based way to improve the recall is to use rules to simplify the non-standard words, e.g. by reducing the number of character repetitions to a maximum of 3 (Han and Baldwin, 2011) thereby detecting spelling variants like {*loool*, *looooooooool*}.

Another difference between contemporary CMC texts and historical data is the amount of text that is available. Therefore, e.g., using context representations should give better results for this type of data than for the GML data.

In this paper, spelling-variant detection has been approached on the type level. However, there is variation on the token level as well (Jurish, 2010b). For example, the dative singular of the name *Maria* appears as *maria* and as *marien* in the GML data, whereas the nominative singular only appears as *maria*. Therefore, *marien* is a spelling variant of *maria* in *do he comen was to maria* 'when he came to Maria', but not a spelling variant for *maria* in *maria hett gesegt* 'Maria said'. One way to approach spelling-variant detection on the token level is to rank spelling variants generated for the type. One possible starting point for this is the system presented by Roark and Sproat (2014) to expand abbreviations. Similarly to the approach presented here, one of their models uses an SVM to evaluate possible expansion candidates. This model also includes features related to the token context, as the abbreviation expansion is done for specific tokens.

Furthermore, for the experiments presented here, we used texts that have been tokenized manually. This removed spelling variation that involves white space. E.g. the GML word for 'kingdom' appears as *koninckryke*, *konnick ryke* and *konynck ryke* in the texts. In order to detect this kind of variation, tokenization has to be combined with spelling-variant detection or spelling-variant detection has to be extended to token n-grams.

5.3 Applications

Apart from an intrinsic evaluation of spelling-variant detection as in this paper, we will also evaluate it extrinsically. Next to the approaches that use detected spelling variants to improve the accuracy of POS tagging and lemmatization, we will employ spelling variants in other tasks.

One of these tasks is normalization. While we presented spelling-variant detection as an alternative to normalization in the absence of an existing standard, it should also be usable to complement normalization as normalization has to deal with spelling variation in non-standard words while mapping these to standard words. For instance, Jin (2015) uses an approach for normalization, where for non-standard words, firstly, normalization candidates are generated, and, secondly, the most probable of these candidates is selected. The candidate generation used in the original approach cannot generate the correct candidate for spelling variants of non-standard words that did not appear in the training data, e.g., *you are* as a normalization candidate for *urr* will not be generated if only *ur* as a non-standard variant of *you are* is known from the training data. Similarly to the approach that Barteld et al. (2016) used to improve the lemmatization of non-standard texts, the knowledge that *urr* is a spelling variant of *ur* could be used to generate the candidate *you are* and thereby improve the coverage of the generator.

Detected spelling variants could also be used as the basis for an artifical standard that can then be used as the target for normalization, where no standard exists. A simple first approach for this would be to transform the spelling variant relation into a clustering by using the symmetric transitive closure and take the most frequent form for each cluster as the standard form.

Another use case that we are interested in is the detection of annotation errors in a corpus. One approach to do this is to use variation n-grams (Dickinson and Meurers, 2003) to detect potential errors. In this approach, variation in the annotation of identical n-grams is used to detect annotation errors. Spelling variation, however, leads to the situation that two identical n-grams on the level of the morphological word appear as different n-grams on the observable type level (cf. Example 2) which affects the recall of this approach. We want to employ spelling-variant detection to identify n-grams that are identical on the level of the morphological word but differ on the type level before employing the variation n-gram method for annotation error detection.

Acknowledgements

This work has been supported by the German Research Foundation (DFG), grant SCHR 999/5-2. I would like to thank the anonymous reviewers for their helpful remarks.

References

Sreangsu Acharyya, Sumit Negi, L. V. Subramaniam, and Shourya Roy. 2009. Language independent unsupervised learning of short message service dialect. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3):175–184.

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135. Association for Computational Linguistics.

Alistair Baron, Paul Rayson, and Dawn Archer. 2009. Word frequency and key word statistics in historical corpus linguistics. *Anglistik: International Journal of English Studies*, 20(1):41–67.

Fabian Barteld, Ingrid Schröder, and Heike Zinsmeister. 2015. Unsupervised regularization of historical texts for POS tagging. In *Proceedings of the Workshop on Corpus-Based Research in the Humanities (CRH)*, pages 3–12.

Fabian Barteld, Ingrid Schröder, and Heike Zinsmeister. 2016. Dealing with word-internal modification and spelling variation in data-driven lemmatization. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 52–62. Association for Computational Linguistics.

Marcel Bollmann, Stefanie Dipper, Julia Krasselt, and Florian Petran. 2012. Manual and Semi-automatic Normalization of Historical Spelling—Case Studies from Early New High German. In *Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012), LThist 2012 Workshop*, pages 342–350.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.

Maria Alina Ciobanu and Liviu P. Dinu. 2014. Automatic detection of cognates using orthographic alignment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 99–105. Association for Computational Linguistics.

Maria Alina Ciobanu and Liviu P. Dinu. 2015. Automatic discrimination between cognates and borrowings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 431–437. Association for Computational Linguistics.

Markus Dickinson and Detmar W. Meurers. 2003. Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 107–114.

Stefanie Dipper. 2011. Morphological and Part-of-Speech Tagging of Historical Language Data: A Comparison. *Journal for Language Technology and Computational Linguistics (JLCL)*, 26(2):25–37.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369. Association for Computational Linguistics.

Gerd Fritz. 2006. *Historische Semantik*. Metzler, Stuttgart and others, 2nd edition.

Phani Gadde, L. V. Subramaniam, and Tanveer A. Faruquie. 2011. Adapting a WSJ trained part-of-speech tagger to noisy text: Preliminary results. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, pages 5:1–5:8.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 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*, pages 42–47. Association for Computational Linguistics.

Luís Gomes and José G. P. Lopes. 2011. Measuring Spelling Similarity for Cognate Identification. In Luis Antunes and H. Sofia Pinto, editors, *Progress in Artificial Intelligence*, Lecture Notes in Computer Science 7026, pages 624–633. Springer, Berlin, Heidelberg.

Stephan Gouws, Dirk Hovy, and Donald Metzler. 2011. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90. Association for Computational Linguistics.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378. Association for Computational Linguistics.

Andreas W. Hauser and Klaus U. Schulz. 2007. Unsupervised learning of edit distance weights for retrieving historical spelling variations. In *Proceedings of the First Workshop on Finite-State Techniques and Approximate Search*, pages 1–6.

Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska De Jong, and Uzay Kaymak. 2015. Exploiting Emoticons in Polarity Classification of Text. *Journal of Web Engineering*, 14(1-2):22–40.

Ning Jin. 2015. Ncsu-sas-ning: Candidate generation and feature engineering for supervised lexical normalization. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 87–92. Association for Computational Linguistics.

Bryan Jurish. 2010a. Comparing canonicalizations of historical german text. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 72–77. Association for Computational Linguistics.

Bryan Jurish. 2010b. More than Words: Using Token Context to Improve Canonicalization of Historical German. *Journal for Language Technology and Computational Linguistics (JLCL)*, 25(1):23–39.

Mike Kestemont, Walter Daelemans, and Guy De Pauw. 2010. Weigh your words—memory-based lemmatization for Middle Dutch. *Literary and Linguistic Computing*, 25(3):287–301.

Catherine Kobus, François Yvon, and Géraldine Damnati. 2008. Normalizing sms: are two metaphors better than one ? In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 441–448. Coling 2008 Organizing Committee.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association of Computational Linguistics*, 3:211–225.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. 2014. Standardizing Tweets with Character-Level Machine Translation. In Alexander Gelbukh, editor, *15th International Conference CICLing 2014, Proceedings, Part II*, Lecture Notes in Computer Science 8404, pages 164–175. Springer, Berlin, Heidelberg.

Pavel Logačev, Katrin Goldschmidt, and Ulrike Demske. 2014. POS-tagging historical corpora: The case of Early New High German. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT-13)*, pages 103–112.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR) 2013, Workshop Track*.

Hermann Niebaum. 2000. Phonetik und Phonologie, Graphetik und Graphemik des Mittelniederdeutschen. In *Sprachgeschichte. Ein Handbuch zur Geschichte der deutschen Sprache und ihrer Erforschung*. De Gruyter, Berlin, Boston, 2nd edition.

Robert Peters and Norbert Nagel. 2014. Das digitale ‚Referenzkorpus Mittelniederdeutsch / Niederrheinisch (ReN)‘. *Jahrbuch für Germanistische Sprachgeschichte*, 5(1):165–175.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Synthesis Lectures on Human Language Technologies 17. Morgan & Claypool Publishers.

Martin Reynaert. 2009. Parallel identification of the spelling variants in corpora. In *Proceedings of the Third Workshop on Analytics for Noisy Unstructured Text Data 2009*, pages 77–84.

Martin Riedl and Chris Biemann. 2013. Scaling to large3 data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 884–890. Association for Computational Linguistics.

Brian Roark and Richard Sproat. 2014. Hippocratic abbreviation expansion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369. Association for Computational Linguistics.

Yves Scherrer and Tomaž Erjavec. 2016. Modernising historical Slovene words. *Natural Language Engineering*, 22(6):881–905.

Klaus Schulz and Stoyan Mihov. 2002. Fast string correction with Levenshtein-automata. *International Journal of Document Analysis and Recognition*, 5:67–85.

Irina Sergienya and Hinrich Schütze. 2015. Learning better embeddings for rare words using distributional representations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 280–285. Association for Computational Linguistics.

Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Ian H. Witten, Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Burlington, MA, 3rd edition.

Yi Yang and Jacob Eisenstein. 2016. Part-of-speech tagging for historical english. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1328. Association for Computational Linguistics.

Replication issues in syntax-based aspect extraction for opinion mining

Edison Marrese-Taylor, Yutaka Matsuo
Department of Technology Management for Innovation
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
emarrese,matsuo@weblab.t.u-tokyo.ac.jp

Abstract

Reproducing experiments is an important instrument to validate previous work and build upon existing approaches. It has been tackled numerous times in different areas of science. In this paper, we introduce an empirical replicability study of three well-known algorithms for syntactic centric aspect-based opinion mining. We show that reproducing results continues to be a difficult endeavor, mainly due to the lack of details regarding preprocessing and parameter setting, as well as due to the absence of available implementations that clarify these details. We consider these are important threats to validity of the research on the field, specifically when compared to other problems in NLP where public datasets and code availability are critical validity components. We conclude by encouraging code-based research, which we think has a key role in helping researchers to understand the meaning of the state-of-the-art better and to generate continuous advances.

1 Introduction

Aspect-based opinion mining is one of the main frameworks for sentiment analysis. It aims to extract fine-grained opinion targets from opinion texts and its importance resides in the fact that without knowing the aspects, opinion analyses are of limited use (Liu, 2012). The concept originated more than 10 years ago as a specific case of sentiment analysis and has gradually gained relevance as a concrete and complete problem in opinion mining. The key task in aspect-based sentiment analysis is to extract the aspects or targets that have been commented in opinion documents. Sentiment orientation can be obtained later based on the extracted terms using or adapting any of the generic approaches for sentiment classification. Therefore, an important amount of focus has been posed on the problem of aspect extraction.

Researchers have proposed several methods for aspect extraction so far, and many authors consider that these approaches largely fall into three main categories. On the one hand, we find syntactical or linguistic methods, which are generally based on other basic NLP tasks, such as POS-tagging and parsing, plus some fixed rules or rankings mechanisms. On the other hand, we find purely statistical approaches, which are mainly based on topic modeling. Finally, we also find extensive work on supervised learning methods, in which case the problem is approached as sequence labeling. Experiments with both Neural Networks and Graphical Models have reported fairly good results so far.

A review of the literature in syntactical approaches showed us that most of the proposed ideas are inspired by or directly built on top of previous methods. Papers generally include detailed comparisons of the approaches, but we found very few publications accompanied by code releases that make it easier to effectively compare and contrast methods. We believe the lack of code availability is increasingly becoming a threat to validity in the field by adding layers of obscurity to new approaches, specially to those that are built on top of previous ideas.

Given the current state in the field, in this paper we study replicability issues in aspect-based opinion mining. We focus on syntactic methods, which tend to show a lower degree of transparency due to the increasing level of model complexity and the lack of code availability. In that sense, in this work we want to encourage discussion on this topic by addressing some key questions.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 23–32, Valencia, Spain, April 3-7 2017.

1. Are the explanations given in the papers generally sufficient to replicate the proposed models?

2. Do differences in preprocessing have a big impact on performance?

3. Do parameters need to be heavily tuned in order to achieve the reported performance?

Our goal throughout this paper is to start exploring possible answers to these questions and provide an environment for further discussions and improvements. We will try to tackle the questions keeping in mind that reproducibility of an experimental result is a fundamental assumption in science. As we will see in the next section, the inability to replicate the experimental results published in a paper is an issue that has been considered in various other machine learning and computer science conferences. There have been several discussions arising from this issue and there seems to be a widespread view that we need to do something to address this problem. We would like to join this quest too.

2 Related Work

Aspect-based opinion mining aims to identify the aspects of entities being reviewed in an text and to determine the sentiment reviewers express for each aspect. Aspects usually correspond to arbitrary topics considered important or representative of the text that is being analyzed.

The aspect-based approach has become fairly popular. Since its conception, arguably after Hu and Liu (2004b), many unsupervised approaches based on statistical and syntax analysis such as Qiu et al. (2011) and Liu et al. (2012) have been developed. While here we specifically tackle these kind of models, other popular unsupervised techniques such as Mukherjee and Liu (2012) are based on LDA.

On the other hand, existing supervised approaches in the field are mainly based on sequence labeling. Since 2014 the SemEval workshop included a shared task on the topic (Pontiki et al., 2014), which has also encouraged the development of new supervised methods. We find approaches based on CRFs such as Mitchell et al. (2013) and deep learning (Irsoy and Cardie, 2014) (Liu et al., 2015a), (Zhang et al., 2015).

The replicability issue has been tackled numerous times in different areas of science. For example, Casadevall and Fang (2010) explore the importance and limits of reproducibility in scientific manuscripts in the field of Microbiology. In the field of Machine Learning, Drummond (2009) discusses issues arising from the inability to replicate the experimental results published in a paper. Also, Raeder et al. (2010) show that when comparing the performance of different techniques some methodological choices can have a significant impact on the conclusions of any study.

Furthermore, we also find studies in Software Engineering. For example, Monperrus (2014) aimed to contribute to the field with a clarification of the essential ideas behind automatic software repair and included an in-depth critical analysis of Kim et al. (2013), an approach that had been published the year before in the same conference.

It is also possible to find work on replicability in Natural Language Processing. Conferences such as CICLing have undertaken maximum effort —though so far rather fruitless— in order to address the topic, giving a special prize every year to the best replicable paper[1]. In addition, the proceedings of the ACL conference have included words on this topic on several occasions. For example, Kilgarriff (2007) introduces the issues of data cleaning and pre-processing, specially for those cases that involve crawling and/or downloading linguistic data. The paper claims that even though expertise and tools are available for most of these preprocessing steps, such as lemmatizers and POS-taggers for many languages, in the middle there is a logjam and questions always arise. The authors indicate that it seems to be undeniable that even though cleaning is a low-level, unglamorous task, it is yet crucial: the better it is done, the better the outcomes. All further layers of linguistic processing depend on the cleanliness of the data.

On the other hand, Pedersen (2008) presents the sad tale of the Zigglebottom tagger, a fictional tagger with spectacular results. However, the code is not available and a new implementation does not yield the same results. In the end, the newly implemented Zigglebottom tagger is not used, because it does not lead to the promised results and all effort went to waste. Fokkens et al. (2013) go further and actually experiment with what they

[1] http://cicling.org/why_verify.htm

think may be a real-world case of the Zigglebottom tagger, particularly, with the NER approach by Freire et al. (2012). The reimplementation process involved choices about seemingly small details such as rounding to how many decimals, how to tokenize or how much data cleanup to perform. They also tried different parameter combinations for feature generation, but the algorithm never yielded the exact same results. Particularly, their best run of their first reproduction attempt achieved nearly a 20% drop in F-measure on average. Other authors such as Dashtipour et al. (2016) have worked on the same issue but for the task of sentiment classification, being unable to replicate the results of several papers. Our work is directly related to these since here we also attempt to reimplement other approaches.

3 Empirical Replication Study

As a first step, we first devoted ourselves to creating a friendly environment for experimentation. The goals of developing this framework were the following. (a) To provide a public Python implementation of notable algorithms for aspect extraction in aspect-based opinion mining that to date lack available implementations, (b) To provide an implementation that is easy to extend and thus to allow researchers to build novel approaches based on the routines we provide, and (c) To increase transparency in the field by providing full details about preprocessing steps, parameter setting and model evaluation. We are publicly releasing our code in GitHub[2], so it will welcome bug fixes, extensions and peer validation of its contents.

Our framework is an object-oriented package that is centered on the representation of a sentence as a property-rich object. Likewise, sentences are composed of tokens, which represent words and other punctuation marks with their respective properties such as stems, POS-tags, IOB-tags for chunks and dependency relation tags, among others. We have also developed wrappers for some popular packages for NLP, concretely the Stanford CoreNLP and Senna. This allows us to easily experiment with different tokenizers, stemmers, POS taggers, chunkers and parsers.

Our package also includes a module for corpora management, which provides easy access to the set of linguistic resources needed. We include parsers for word lists such as stopwords, opinion lexicons and also for more complex data structures regarding aspect-based opinion mining. In particular, we work with the well-known *Customer Review Dataset* (Hu and Liu, 2004a; Hu and Liu, 2004b) which became the de facto benchmark for evaluation in syntax-based aspect-based opinion mining. This is also a very important part of our environment.

We also include a simple module devoted to model evaluation, which makes the evaluation process transparent. We see aspect extraction as an information retrieval problem and thus the evaluation is based on precision, recall and F1-score, using exact matching to define a correctly extracted aspect.

On top of the framework we built our implementations of three different aspect extraction techniques, which we selected based on the approach they are based on, their novelty and their importance in the community. As we mentioned earlier, since we limit our study to syntactic approaches, here we explicitly omit algorithms that are intensively based on Web sources —or other private sources or datasets— and also approaches that use topic models or supervised learning models for sequence labeling. We selected three different papers, Hu and Liu (2004b), Qiu et al. (2011), Liu et al. (2012). In the subsections below, we proceed to comment on the reasons for each choice and give details on our implementations.

3.1 Frequency-Based Algorithm (FBA)

We first consider the aspect extraction algorithm by Hu and Liu (2004b), which pioneered on the problem of aspect-based opinion mining. This work is still being considered as a baseline for comparison and contrast with new approaches by most of the work on syntactic approaches in the literature. Despite this, there seems to be no available implementation of this technique to the best of our knowledge. These were our main motivations to work with this technique.

The aspect extraction procedure is based on frequent itemset mining, which given a database of transactions and a minimum support threshold min_{sup} extracts the set of all the itemsets appearing in at least min_{sup} transactions —an itemset is just an unordered set of items in a transaction. In this case, each transaction is built using the nouns and words in the noun phrases of a sentence. Later, stopword removal, stemming

[2] `github.com/epochx/opminreplicability`

and fuzzy matching are applied to the transactions in order to reduce the term dimensionality and to deal with word misspellings. Authors do not mention which stemming algorithm they use, so we resort both the well-known Porter stemmer and the Stanford lemmatizer, which can be regarded as the standard choices.

Regarding fuzzy matching, the approach uses Jokinen and Ukkonen (1991), but authors simply state that [... *all the occurrences of "autofocus" are replaced with "auto-focus"*]. This description was insufficient to give us a full notion of how the process is carried out, specially since arbitrary word replacements can have an important impact when extracting aspects based on their frequency.

Similarly to de Amorim and Zampieri (2013), who proposed a clustering method for spell checking, here we use clustering with the Levenshtein distance ratio as similarity metric to group terms. We tried with different strategies of hierarchical clustering and, based on our exploratory experiments, we decided to use *complete linkage* to extract flat clusters so that the original observations in each flat cluster have a maximum cophenetic distance given by a parameter min_{sim}. Finally, we represent each stem as a fixed single stem in its cluster, keeping an index back from each of the original unstemmed words to its cluster, so we are later able to recover the terms as they appeared originally.

Authors later proceeded to mine *frequent* occurring phrases by running the association rule miner CBA (Liu et al., 1998), which is based on the Apriori algorithm. The paper indicates that the Apriori algorithm works in two steps, first finding all frequent itemsets to later generate association rules from the discovered itemsets, so authors state they only needed the first step and use the CBA library for this part. This seems reasonable since it is a known fact that it is very efficient to use frequent itemsets to generate association rules (Agrawal et al., 1993). They limited itemsets to have a maximum of three words as they believed that a product feature contained no more than that number of terms. For minimum support, they defined an itemset as *frequent* if it appeared in more than 1% of the review sentences. In our case, since the CBA library was never released, we resort to an open-source implementation of the Apriori algorithm for frequent itemset mining (Borgelt, 2012; Pudi and Haritsa, 2002; Pudi and Haritsa, 2003).

After itemset mining, two pruning steps are applied in order to get rid of the incorrect, uninteresting and redundant features. We implemented these pruning techniques closely following the details given in the paper.

Finally, extracted aspects are used to extract infrequent features that might also be important. In order to do so, they used terms in a lexicon as pivots to extract those nearby nouns that the terms modify. To generate the list of opinion words, they extracted the nearby adjective that modifies each feature on each of the sentences in which it appears, using stemming and fuzzy matching to take care of word variants and misspellings. The paper states that *"a nearby adjective refers to the adjacent adjective that modifies the noun/noun phrase that is a frequent feature"*. However, it is not clear how they really find these adjectives. In our implementation, we defined a distance window from the aspect position and extract all adjectives that appear within this window. The size of this window became another parameter of the model. Finally, to extract infrequent features, authors checked those sentences that contain no frequent features but one or more opinion words and then extracted the nearest noun/noun phrase.

We try to keep parameter setting as close as possible to the values reported by the original paper, but for POS-tagging we use CoreNLP or Senna instead of NLProcessor 2000. To obtain flat noun phrases, we use the Penn Treebank II output generated by the Stanford Constituency Parser and apply the same Perl script[3] used to generate the data for the CoNLL-2000 Shared Task.

3.2 Dependency-Based Algorithm (DBA)

Our second implemented model is Double Propagation (Qiu et al., 2011), an approach that is fundamentally based on dependency relations between words for both aspect/target and opinion word extraction. This paper pioneered on the usage of dependency grammars to extract terms by iteratively using a set of eight rules based on dep-relations and POS-tags. Basically, the process starts with a set of *seed* opinion words whose orientation is already known. In general, this is a reasonable assumption since several opinion lexicons already exist in the literature. The seeds are firstly used to extract aspects, which are defined as nouns that

[3] http://ilk.uvt.nl/team/sabine/homepage/software.html

are modified by the seeds. Aspects are later used to extract more opinion words indicated by adjectives, other aspects and so on. This iterative process that propagates the knowledge with the help of the rules ends when no more opinion words or aspects are extracted.

In the original paper, the set of dependency relationships given by the MINIPAR parser (Lin, 2003) is used to develop the word extraction rules. We actually could not find the binaries on-line since the official website[4] is down; other binaries found on the Web were corrupted and unusable. This convinced us that MINIPAR can be regarded as a rather outdated model, so we decided to use the Stanford Parser Manning et al. (2014) instead, which is among state-of-the-art models in the field. Our choice is supported by the results of Liu et al. (2015b), who successfully work with Double Propagation based on the Stanford dependency parser. Since the Stanford dep-tags differ from the tagset used by MINIPAR, we use the equivalences defined in the aforementioned paper.

After the extraction steps, the approach proceeds to apply a clause pruning phase. For each clause on each sentence, if it has more than one aspect and these are not connected by a conjunction, only the most frequent one is kept. In the paper, authors simply state that they [*"identify the boundary of a clause using MINIPAR"]* and do not explain how to determine if the aspects are connected by the conjunction. We identify clauses by finding the set of non-overlapping parse sub-trees with label "S". To determine if the aspects are connected by any existing conjunction in a sentence, we simply check if the conjunction appears between the aspects in the same clause.

The next step was to prune aspects that may be names of other products or names of product dealers, which may appear due to comparisons. In this case, the procedure is based on pre-defined patterns which are first matched in the text to later check if nearby nouns had previously been extracted as aspects. These are removed. The definition of *nearby noun* is not given in the paper, so we add it as another parameter for the model, again using the notion of distance windows.

Finally, a rule is proposed to identify aspect phrases by combining each aspect with Q consecutive nouns right before and after the aspect and K adjectives before it. After obtaining the aspect phrases, another frequency-based pruning is conducted, removing aspects that appear only once in the dataset. Again, here we tried to set all the parameters as reported by the authors. Based on preliminary experiments, we decided to also eliminate those terms that were extracted by leveraging on aspects that were later pruned, since they may introduce noise.

[4] https://webdocs.cs.ualberta.ca/~lindek/minipar.htm

3.3 Translation-based Algorithm (TBA)

The work of Liu et al. (2012) is a novel application of classic statistical translation models and Graph Theory to extract opinion targets. Novelty and the good results obtained by the approach were our main motivations to work with this paper.

For target extraction, the authors proposed a technique based on statistical word alignment. Specifically, they used a constrained version of the well-known IBM statistical alignment models (Brown et al., 1993). The proposal is directly related to monolingual alignment, as proposed by Liu et al. (2009). For monolingual alignment, the parallel corpora fed to the model is simply two copies of the same corpus. At the same time, the condition that words cannot be aligned to themselves is added. Liu et al. (2012) still use a monolingual parallel corpus but set the constraint that nouns and noun phrases can only be aligned to adjectives or vice-versa, meanwhile the rest of the words can be aligned freely. As a result, authors are able to capture noun/noun phrase-adjective relations that have longer spans that direct dependency relations in a sentence.

Since the IBM alignment models work at word granularity and then need to receive tokenized sentences as input, here we assume that authors first proceeded to group noun phrases in single tokens. According to the paper, they resorted to the *C-value* technique (Frantzi et al., 2000) for multi-word term extraction, which was originally developed to detect technical terminology in medical documents, but was also previously used in the domain of opinion mining by Zhu et al. (2009). The method firstly generates a list of all possible multi-word terms and later ranks them using statistical features from the corpus. Even though in the original paper candidate multi-word terms are extracted using fixed patterns, authors decided to generate all candidates as simple n-grams (with $max_n = 4$). We implemented and experimented

with both fixed pattern and the n-gram versions for the *C-value* technique. We also added a simple heuristic that works without ranking, grouping sets of nouns and other related figures that appear on the same parse NP sub-tree.

After the most likely constrained alignments are obtained for each sentence, authors estimated noun/noun phrase-adjective pair associations as the harmonic mean between the mutual translation probabilities. Finally, they built a bipartite graph with the words and estimate the confidence of each target candidate being a real target using the mined associations and applying a graph-based algorithm based on random walking. This is basically an iterative algorithm that exploits the mutual reinforcement between terms as given by the associations. Authors set the relevance of each target as the initial value of confidence, defining relevance as the normalized *tf-idf* scores of the candidates, where *tf* is the frequency on each term in the corpus and *idf* is the inverse document frequency obtained using the Google N-gram Corpus.

In the paper, authors experimented with IBM1-3 models and showed that fertility parameters introduced by the third model help to improve the performance by a small margin. The estimation of this model is rather complicated, in this case specially since it also includes a constrained version of the hill-climbing heuristic, so in our implementation we only include our versions of the IBM1-2 models. Regarding parameters, we set the proportion of candidate importance $\lambda = 0.3$ and the maximum series power parameter $k = 100$, as given by the original paper. To compute the initial relevance of each candidate, authors use the Google Ngram corpus to obtain the *idf* of a term. Due to the lack of explanations on what they consider as a document, we resorted to the the English Wikipedia. To calculate the *idf* score of a term, we count the number of articles that contain the queried term and compare it to the total number of articles. When we find no articles for a given term, we simple use a minimum article count of 1.

Finally, the authors stated that the targets with higher confidence scores than a certain threshold t are extracted as the opinion targets, but they do not specify the value they use. We let our implementation output the unfiltered list of candidates and their confidences and find the best value of the threshold later.

4 Preliminary Results

As we have shown, the implementation process involved choices about several details that were not clear or not mentioned on the papers. In our experiments we have found that even when trying different parameter combinations we remain unable to yield the exact same results in the original papers. Below we summarize our best results and findings for each algorithm.

Corpus	Original		Ours	
	P	R	P	R
Apex DVD Player	0.797	0.743	0.389	0.355
Creative MP3 Player	0.818	0.692	0.293	0.319
Nikon Camera	0.792	0.710	0.265	0.457
Nokia Phone	0.761	0.718	0.328	0.489
Canon Camera	0.822	0.747	0.352	0.286
Average	0.8	0.72	0.325	0.381

Table 1: Performance comparison for FBA.

Table 1 compares our implementation's best results so far with the values reported by Hu and Liu (2004b). We remain unable to replicate the performance reported by the authors and see a big drop for both precision and recall in all the datasets. In our experiments, we noted that the most sensitive parameter was min_{sup} for itemset mining. We also experimented omitting the pruning steps and observed that precision and recall were not too different from the results we obtained with pruning.

We also observed that several parameters configurations conveyed the same final performance for each corpus. Among the 1470 per-corpus parameter configurations we tried, we found 18 optimal settings for both the Apex DVD Player and Canon Camera corpora, 16 for Nikon Camera, 6 for Creative MP3 Player and 3 for Nokia Phone.

Differences in preprocessing did not offer consistent differences in performance. For the Apex DVD Player, Creative MP3 Player and Canon Camera corpora we found that processing with SennaConstParser + CoreNLPDepParser conveys the best results. For the Nikon Camera corpus, adding the PorterStemmer to the latter gave us the best performance. For the case of the Nokia Phone corpus, the pipeline CoreNLPDepParser + CoNLL2000Chunker gave us the best results.

In the original paper, authors reported the performance of the model at different stages, showing that average values of precision an recall for the itemset mining stage are 0.68 and 0.56 respectively. We were surprised to find out that we could

not even replicate these results, specially considering that only two parameters are at play at this level. As shown by the original paper, the final performance achieved is actually mainly due to the output of the itemset mining phase. We believe this is the reason why we observed some parameters have minimum impact on the performance. This means that no matter how good the pruning strategies are, results will not be as good as the original if we remain unable to replicate the output of the itemset mining phase.

Corpus	Original		Ours	
	P	R	P	R
Apex DVD Player	0.90	0.86	0.239	0.328
Creative MP3 Player	0.81	0.84	0.180	0.319
Nikon Camera	0.87	0.81	0.194	0.287
Nokia Phone	0.92	0.86	0.287	0.359
Canon Camera	0.90	0.81	0.201	0.356
Average	0.88	0.83	0.220	0.330

Table 2: Performance comparison for DBA.

Regarding DBA, Table 2 summarizes the results we obtained. Again, we see huge differences between our results and the ones reported by the original paper. Moreover, in this case we observe particularly low values for precision. A detailed review of the extracted aspects showed us that in fact many of the extracted terms do not correspond to aspects but rather to common nouns that are not related to the product.

During experimentation, we also added support for different matching strategies —for example, using word stems and including fuzzy matching as in FBA— and although we observed improvements on the results, these were marginal. We used different opinion word seeds, firstly based only on the words "good" and "bad" and later using 9 same-size subsets of the opinion lexicon provided by Liu[5]. In all cases, our best performing model uses one of these subsets.

As in the previous case, different parameter configurations led to the same performance for each corpus. In this case, among 240 parameter settings for each corpus, we found 12 optimal configurations for the Apex DVD Player corpus and 24 for each the other corpora. Regarding preprocessing, we could not use CoreNLP to transform the constituent trees given by Senna into dep-trees. Constituency trees seemed to be malformed and raised grammar parsing errors, therefore we only experimented using the CoreNLPDepParser + CoNLL2000Chunker pipeline.

[5]http://www.cs.uic.edu/∼liub/FBS/opinion-lexicon-English.rar

Corpus	t^*
Apex DVD Player	160
Creative MP3 Player	200
Nikon Camera	100
Nokia Phone	90
Canon Camera	110

Table 3: Optimal value of t for each corpus.

Table 4 shows a comparison of the results we have obtained so far using our implementation of TBA and the values provided by Liu et al. (2012). Once more, we remain unable to replicate the performance reported by the paper.

On our experiments we tried with all three grouping strategies to generate multi-word terms; namely, our simple heuristic and *C-value* using both n-grams and fixed patterns. We also tried adding a limit for the number of groups generated by the *C-value* technique and used stemming to improve frequency counts. The "ngram" technique turned out to be the best performing on each corpus, although the limit parameter varies from case to case.

As we mentioned earlier, to evaluate the impact of the t parameter whose value was not reported by Liu et al. (2012), we let the model return the unfiltered aspect candidates and evaluate the performance for $t \in [10, 20, ..., t_{max}]$. Note that t_{max} might be different each time. Because of this, the number of parameter configurations we tried for each corpora is slightly different. Instead of reporting each value, we rather report the average of per-corpus evaluated parameter settings, which was 1006. As Table 3 shows, we found that rather than a single cross-corpus optimal value, this parameter needs to be tuned per-corpus. In this sense, when we experimented without setting a threshold we obtained a maximum recall of 0.697 —for the Nokia Phone corpus— but at the cost of precision 0.151. When we set $t = 10$, we obtained a maximum precision of 0.9 but at the cost of recall being lower than five percent. These results mean the model does not seem to generalize well.

Since in our implementation we do not use the IBM3 model, we were aware we could see a difference in the performance. However, based on the results by the original paper, which showed that improvements of IBM3 over IBM2 are small — about 5%— we think it is very unlikely this difference can explain the big drop in performance we

have observed.

Corpus	Original		Ours	
	P	R	P	R
Apex DVD Player	0.89	0.87	0.362	0.389
Creative MP3 Player	0.81	0.85	0.400	0.327
Nikon Camera	0.84	0.85	0.380	0.404
Nokia Phone	0.88	0.89	0.588	0.381
Canon Camera	0.87	0.85	0.400	0.341
Average	0.86	0.86	0.426	0.368

Table 4: Performance comparison for TBA.

5 Discussion and further directions

The ongoing empirical study we introduce in this paper has provided concrete cases to help us answer the questions that motivate this paper. As seen, we have so far failed to reproduce the original results in the three studied cases. Even though several reasons may be the cause for this failure, we think further experimentation can allow us to determine the key elements that would explain the differences. In fact, our preliminary experiments have already helped us isolate specific parameters for each model that seem to more strongly improve the performance. Our results show that parameters that are closely related to the core of the extraction methods, such as min_{sup} for FBA and the confidence threshold t for TBA seem in fact to be playing these key roles.

We are planning to run controlled experiments in order to isolate as much as possible the effect of each parameter or processing step and understand their interplay. This will enable us to tell where important implementation differences between our version and the original version may be. Given that we do not have access to the original codes, it is only by means of these inferred differences that we can gain real insights on where the keys for replicability lay.

We believe the results in this paper already prove that explanations given in the original papers were generally insufficient to correctly replicate the models. The lack of resources —except for the evaluation datasets— caused us to navigate in the dark as we could not reverse-engineer many intermediate steps. Certain details of preprocessing and parameter setting are only mentioned superficially or not at all in the original papers. However, many of these seemingly small details did make a big difference in our results. We understand there is often not enough space in the manuscripts to capture all details, specially since they are generally not the core of the research described. However, code releases play a critical role in uncovering these details and making research at least replicable.

Regarding pre-processing, in our experiments so far with both Senna and CoreNLP we saw performance differences that are however not consistent, which seems to indicate that there is no optimal preprocessing pipeline for each algorithm. On the other hand, model parameters do not seem to be correlated with pre-processing choices, although we did find a single case in which a special pre-processing step lead to better results in a single corpus.

Though we could not replicate the results published in the original papers, we have shown that parameter values as reported by these papers do not necessarily yield the best results. Moreover, parameters that may seem unimportant turned out to cause important performance differences for us. Most parameters indeed had to be heavily tuned in order to achieve the best performance.

Finally, the poor results obtained by our implementations also leave us puzzled about how the evaluation is really performed on the original papers. Authors do not give much details on this topic. For example, (Hu and Liu, 2004b) use stemming and simply eliminate some words from the text based on their fuzzy matching approach. This means their extracted terms are word stems only. However, we do not know if stemming is also applied to the gold standard to evaluate. We manually examined the *Customer Review Dataset* and discovered that the manually extracted aspects do not seem to be stemmed. Moreover, we noted several inconsistencies in the annotation. This issue raises more questions for our research.

6 Conclusions

We have presented three replication cases in the domain of aspect-based opinion mining and shown that repeating experiments in the field is a complex issue. The experiments we designed and carried out have helped us answer our research questions, also raising some new ones. These answers seem to indicate that explanations on pre-processing, models specifications and parameter setting are generally insufficient to successfully replicate papers in the field.

Our observations indicate that sharing data and software play key roles in allowing researchers to

completely understand how methods work. Sharing is key to facilitating reuse, even if the code is imperfect and contains hacks and possibly bugs. Having access to such a set-up allows other researchers to validate research and to systematically test the approach in order to learn its limitations and strengths, ultimately allowing to improve on it.

References

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pages 207–216, New York, NY, USA. ACM.

Christian Borgelt. 2012. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):437–456.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

A. Casadevall and F. C. Fang. 2010. Reproducible Science. *Infection and Immunity*, 78(12):4972–4975, December.

Kia Dashtipour, Soujanya Poria, Amir Hussain, Erik Cambria, Ahmad Y. A. Hawalah, Alexander Gelbukh, and Qiang Zhou. 2016. Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques. *Cognitive Computation*, 8(4):757–771.

Renato Cordeiro de Amorim and Marcos Zampieri. 2013. Effective Spell Checking Methods Using Clustering Algorithms. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 172–178, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Chris Drummond. 2009. Replicability is not reproducibility: nor is it good science. In *Proceedings of the Evaluation Methods for Machine Learning Workshop*, Montreal, Canada.

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from Reproduction Problems: What Replication Failure Teaches Us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701, Sofia, Bulgaria, August. Association for Computational Linguistics.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130.

Nuno Freire, José Borbinha, and Pável Calado, 2012. *The Semantic Web: Research and Applications: 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, chapter An Approach for Named Entity Recognition in Poorly Structured Data, pages 718–732. Springer Berlin Heidelberg, Berlin, Heidelberg.

Minqing Hu and Bing Liu. 2004a. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Minqing Hu and Bing Liu. 2004b. Mining Opinion Features in Customer Reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 755–760, San Jose, California. AAAI Press.

Ozan Irsoy and Claire Cardie. 2014. Opinion Mining with Deep Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728, Doha, Qatar, October. Association for Computational Linguistics.

Petteri Jokinen and Esko Ukkonen. 1991. Two algorithms for approxmate string matching in static texts. In *International Symposium on Mathematical Foundations of Computer Science*, pages 240–248. Springer.

Adam Kilgarriff. 2007. Googleology is bad science. *Computational linguistics*, 33(1):147–151.

Dongsun Kim, Jaechang Nam, Jaewoo Song, and Sunghun Kim. 2013. Automatic patch generation learned from human-written patches. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 802–811. IEEE Press.

Dekang Lin. 2003. Dependency-Based Evaluation of Minipar. In Anne Abeill, editor, *Treebanks*, volume 20 of *Text, Speech and Language Technology*, pages 317–329. Springer Netherlands.

Bing Liu, Wynne Hsu, and Yiming Ma. 1998. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*.

Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. 2009. Collocation extraction using monolingual word alignment method. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 487–495. Association for Computational Linguistics.

Kang Liu, Liheng Xu, and Jun Zhao. 2012. Opinion Target Extraction Using Word-Based Translation Model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language*

Processing and Computational Natural Language Learning, pages 1346–1356, Jeju Island, Korea, July. Association for Computational Linguistics.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015a. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, Lisbon, Portugal, September. Association for Computational Linguistics.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015b. Automated Rule Selection for Aspect Extraction in Opinion Mining. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1291–1297, Buenos Aires, Argentina. AAAI Press.

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open Domain Targeted Sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654, Seattle, Washington, USA, October. Association for Computational Linguistics.

Martin Monperrus. 2014. A critical review of automatic patch generation learned from human-written patches: essay on the problem statement and the evaluation of automatic software repair. In *Proceedings of the 36th International Conference on Software Engineering*, pages 234–242. ACM.

Arjun Mukherjee 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, pages 339–348, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ted Pedersen. 2008. Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Vikram Pudi and Jayant R. Haritsa. 2002. On the Efficiency of Association-Rule Mining Algorithms. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '02, pages 80–91, London, UK, UK. Springer-Verlag.

Vikram Pudi and Jayant R. Haritsa. 2003. ARMOR: Association Rule Mining based on ORacle. In *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI-03)*, volume 90, Melbourne, Florida, USA, November. CEUR Workshop Proceedings.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction Through Double Propagation. *Computational Linguistics*, 37(1):9–27, March.

T. Raeder, T. R. Hoens, and N. V. Chawla. 2010. Consequences of Variability in Classifier Performance Estimates. In *2010 IEEE International Conference on Data Mining*, pages 421–430.

Meishan Zhang, 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*, pages 612–621, Lisbon, Portugal, September. Association for Computational Linguistics.

Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. 2009. Multi-aspect opinion polling from textual reviews. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1799–1802. ACM.

Discourse Relations and Conjoined VPs: Automated Sense Recognition

Valentina Pyatkin
Department of Computer Science
Sapienza University of Rome
pyatkin@di.uniroma1.it

Bonnie Webber
School of Informatics
University of Edinburgh
bonnie@inf.ed.ac.uk

Abstract

Sense classification of discourse relations is a sub-task of shallow discourse parsing. Discourse relations can occur both across sentences (*inter-sentential*) and within sentences (*intra-sentential*), and more than one discourse relation can hold between the same units. Using a newly available corpus of discourse-annotated intra-sentential conjoined verb phrases, we demonstrate a sequential classification system for their multi-label sense classification. We assess the importance of each feature used in the classification, the feature scope, and what is lost in moving from gold standard manual parses to the output of an off-the-shelf parser.

1 Introduction

Discourse relations can hold between inter-sentential and intra-sentential arguments. As Language Technology has much to gain from recognizing intra-sentential discourse relations (Joty et al., 2015), the Penn Discourse TreeBank project has annotated the discourse senses of conjoined verb phrases in the Wall Street Journal corpus (Webber et al., 2016).

Broadly construed, conjoined VPs are sisters in a parse tree, separated from each other by a conjunction and/or punctuation, and possibly one or more adverbs or adverbial phrases as well. As with other units of discourse, more than one sense relation can hold between conjoined VPs. An explicit conjunction may itself convey multiple senses, or additional senses may arise through inference or be signaled with other lexico-syntactic cues (Webber et al., 2016; Prasad et al., 2014). With no explicit conjunction, sense relations will arise through inference or are signaled with other lexico-syntactic cues. Example (1) illustrates senses arising through inference, even though an explicit connective is also found in the conjunction. Here, 'making the penalties fairer and easier to administer' is the GOAL of 'simplifying the penalties', and the latter is the MANNER of achieving that goal.

(1) Long-debated proposals to *simplify the more than 150 civil penalties* (ARG1) *and make them fairer and easier to administer*(ARG2) are in the House tax bill. [wsj_0293]

Automatic classification of the sense relations that hold between sister VPs can thus be formulated as the following task: given a pair of sister VPs and how they have been conjoined, can the sense relation(s) between them be induced? We have approached this task using two Support Vector Machines in a way that allows multi-label classification. To understand what is contributing to effective classification, we examine the separate contributions of syntactic (Section 4.3) and semantic features (Section 4.4), and then the extent to which information internal to the sister VPs suffices to determine how they relate to one another, or whether features external to the pair are also needed (Section 4.5). We also assess the extent to which performance drops when argument spans are provided by an 'off-the-shelf' parser rather than manual annotation (Section 5).

The novel contribution of this work is its use of multi-label classification in determining the discourse sense(s) that hold between conjoined VPs. This type of sense classification on conjoined VPs has not been done before to our knowledge. The evaluation of the features and the feature scope could provide a useful starting-point for future systems that classify inter-sentential discourse relations. Such a classifier could be in-

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 33–42, Valencia, Spain, April 3-7 2017.

corporated into other NLP systems, such as Machine Translation or Automatic Summarization. Louis et al. (2010), for example, showed the benefit of discourse features as importance indicators for automatic summarization, Meyer et al. (2015) used sense labeled discourse connectives in an improved phrase based machine translation system and Prasad and Joshi (2008) generated questions using properties and arguments of specific discourse relations.

2 Background

The sense annotation of discourse relations is part of shallow discourse parsing, involving the identification of pairs of discourse arguments (*Arg1* and *Arg2*) and the sense(s) i which they are related.

(2) Exxon Corp. *built the plant* (ARG1) **but** *closed it in 1985* (ARG2). [wsj_1748]

Example (2) shows the two arguments and the explicit connective 'but'. The annotators labeled this as expressing both CONCESSION (i.e., closing was not expected) and PRECEDENCE (i.e., closing occurred after building). Discourse relations are signaled either explicitly through a discourse connective, or implicitly, or with some other lexicalization (ALTLEX) such as 'will result in'. In the conjoined VP sub-corpus of the PDTB 3.0 (Webber et al., 2016), the left argument is labeled *Arg1* and the right argument, *Arg2*. The goal of shallow discourse parsing is thus to automatically identify the arguments, their spans, the connective (for an explicit relation), and the sense(s) in which they are related. It is called 'shallow' because it does not recursively construct a discourse 'parse tree' (Stede, 2011). The first end-to-end shallow discourse parsers carried out subtasks in a pipeline, separating the tasks of parsing explicit and implicit discourse relations (Lin et al., 2014; Wang and Lan, 2015).

Shallow discourse parsing of conjoined VPs differs from this model of discourse parsing in that the arguments must be sister VPs in a parse tree. Thus, syntactic parsing (either phrase-structure or dependency) must precede identification of sister VPs, whether there is an explicit connective between them or not. This makes shallow discourse parsing more dependent on parser accuracy than in the past. As we will show in Section 5, parsers often fail to accurately parse conjoined VPs (or conjoined structures in general, (Ficler and Goldberg, 2016)).

In terms of features, Subba and Di Eugenio (2009) mention VerbNet as a resource to generalize the semantics of verbs. Pitler and Nenkova (2009) used a small collection of syntactic features to do single-label sense classification from a set of four high-level sense types. Rutherford and Xue (2014) mention that Brown Clusters are helpful to classify implicit relations. For the machine learning algorithms, Meyer et al. (2015) claim that a Maximum Entropy classifier is suitable for sense classification as it learns feature combinations. Hernault et al. (2010) propose the use of a SVM for its suitability for a larger feature-space.

3 Corpus

The Penn Discourse TreeBank has been extended to cover discourse relations between conjoined VPs occurring in the Penn Wall Street Journal corpus (Webber et al., 2016). Besides this sub-corpus, we are aware of only one corpus of discourse annotated conjoined VPs (Subba and Di Eugenio, 2009). This contains fewer annotated tokens than the current set ($\sim$600, as opposed to $\sim$4600), with several sense labels specific to the instruction domain and with only a single relation able to hold between any two conjuncts.

A total of 4633 conjoined VPs have now been annotated in the PDTB, with 3372 having a single sense and 1261 having multiple senses (Webber et al., 2016). There are three conditions in which multiple sense relations hold between sister VPs[1]:

1. Two Explicit senses: One sense is associated with the explicit conjunction and another with an explicit adverb (e.g. "and later").

2. Explicit and Implicit senses: One sense is associated with the explicit conjunction, while other senses are derived through inference.

3. Explicit and AltLex senses: One sense is associated with the explicit conjunction, while another is expressed through an AltLex (e.g. "at the same time").

The numbers for the three types of multi-label conjunctions can be seen in Table 1, along with the numbers for single-label conjunctions. If there is no explicit connective, the multi-sense relations

[1]There could also have been multiple implicit relations between sister VPs, but none appear in the Conjoined VP subcorpus.

are annotated on a single instance of the conjunction. In cases where one sense comes from the explicit conjunction, while the others are derived through inference, this is implemented as two separate linked tokens, one labeled "Explicit", the other "Implicit". This means that some implicit relations hold between sister VPs with no explicit conjunction between them, and others hold between explicitly-conjoined sister VPs whose additional senses derive through inference. A revised

	single-s.	**multi-s.**
Explicit conjunction	2933	
Explicit adverbial	29	
Implicit (punctuation)	410	
Explicit + Adverbial		214
Explicit + Implicit		1017
Explicit + AltLex		30

Table 1: Single-sense and multi-sense counts.

set of sense labels, consisting of 34 labels, has been used in annotating the Conjoined VP corpus and other recent annotation of the Penn Discourse TreeBank (Webber et al., 2016). The senses of the PDTB are constructed in a hierarchical manner. The first level of the hierarchy distinguishes between 4 different sense categories: TEMPORAL, COMPARISON, CONTINGENCY and EXPANSION (Prasad et al., 2014).

4 Classification

4.1 Baseline

As there currently exists no sense-relation classification system for conjoined VPs, the strongest baseline corresponds to majority properties of the corpus. Different majority classes are attributed to implicit and explicit conjunction. For explicit conjunctions with a connective/adverb, the most common sense per connective/adverb is chosen. For implicit relations the most common implicit sense is selected (TEMPORAL.ASYNCHRONOUS.PRECEDENCE). We apply these rules on the same dataset that is used for the classification approach, with certain senses removed, as will be explained in Section 4.2. The various baselines can be seen in Table 2

4.2 Classification approach

Since several senses occur only rarely in the corpus, while EXPANSION.CONJUNCTION occurs as a sense label on more than 77 % of the tokens, actions had to be taken to avoid optimizing performance by simply learning the majority label. To avoid this false optimization, we only considered senses that occurred at least 30 times in the corpus, and in any given training set, we only allowed up to 500 tokens of EXPANSION.CONJUNCTION. The final sense set used for classification thus consists of the 11 senses in Table 3. Tokens not annotated with at least one of these senses have been removed, and multi-label tokens with only one sense shown in Table 3 have been included as single-label tokens. As a result 2446 conjunctions can be used for training and testing.

A system with two classifiers is used for the multi-label classification task. To prove the effectiveness of this approach, in Section 6 we compare this two-classifier method with another multi-label

	Acc.	Prec.	Rec.	F-m.
Implicit	0.37	0.14	0.37	0.20
Explicit	0.49	0.61	0.49	0.42
Total	0.49	0.58	0.49	0.41

Table 2: Baseline for only implicit relations, only explicit relations and the total dataset.

Comparison	Concession	Arg2-as-denier
Comparison	Contrast	
Contingency	Cause	Result
Contingency	Purpose	Arg2-as-goal
Expansion	Conjunction	
Expansion	Disjunction	
Expansion	Level-of-detail	Arg2-as-detail
Expansion	Manner	Arg1-as-manner
Expansion	Substitution	Arg1-as-subst
Expansion	Substitution	Arg2-as-subst
Temporal	Asynchronous	Precedence

Table 3: The subset of 11 senses used in our classification. The left-hand column shows the high-level category of the relation, and the center column shows mid-level sense category. For senses in which a relation can hold in either direction, the right-hand column specifies which direction holds. In the case of Substitution, both the sense in which Arg1 serves as a substitute for Arg2 (i.e., Arg1-as-subst) and the sense in which Arg2 serves as a substitute for Arg1 (i.e., Arg2-as-subst) are used in classification.

classification approach using a One-Vs-Rest classifier, which employs a separate SVM for every label (Pedregosa et al., 2011). The classification setup can be seen in Figure 1. Two SVM classifiers are trained and tested independently. One classifier, called 'Implicit' classifier, is trained on instances of implicit conjunctions and conjunctions with alternative lexicalizations or discourse adverbials. The 'Explicit' classifier is trained on instances of explicit conjunctions. While relations arising from AltLex or discourse adverbials could technically be seen as explicit conjunctions, we added them to the 'Implicit' classifier's training set for the system to be able to identify multi-label conjunctions containing both an explicit connective and an adverbial/AltLex. As part of the training data, both classifiers are also given negative instances, e.g. training data from the respective other classifier, which the classifier ideally has to label as 'NO'.

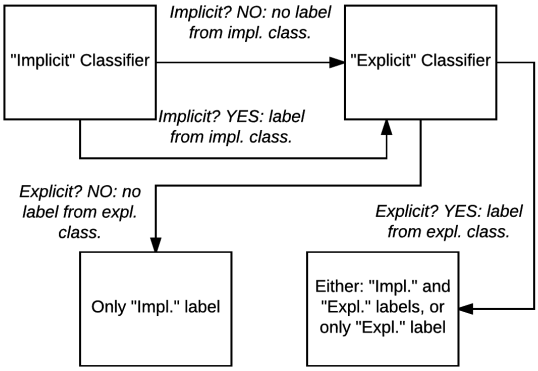


Figure 1: Classification system using two classifiers and negative examples (the order of the two classifiers does not matter as they are independent).

The system starts with both classifiers running in parallel on the same instance. This instance is then assigned an (implicit) sense or is classified as a non-implicit relation by the Implicit classifier and either assigned an (explicit) sense or classified as a non-explicit relation by the Explicit classifier. The order in which the two classifiers are applied is arbitrary, since they operate independently of each other.

After both classifiers finish, their results are combined. The set of the labels from both classifiers, with the NO labels removed, is then the final multi- or single-label instance. This allows for single-label classification, as well as the multi-label cases mentioned in Section 3. A drawback of making both classifiers also predict 'NO' labels is that it could result in both classifiers predicting 'NO', indicating that the system cannot associate any relation to that instance.

As both classifiers learn their parameters independent of the other classifier, the feature selection and evaluation is kept separate for each classifier. The performance of the classifiers is reported using precision, recall and f1-measure. All three measures are calculated for each class separately and then averaged. The f1-measure is also weighted by the number of class-instances, which results in numbers that do not lie between the recall and the precision. The feature analysis is done using a Recursive Feature Elimination algorithm (Pedregosa et al., 2011), which designates weights to the individual features by recursively removing features. For the single-label 'Implicit' and 'Explicit' classifiers the reported measures are obtained using 4-fold cross-validation.

4.3 Syntactic Features

4.3.1 Experiments

Since the connective and its sense-dependent distribution are used in the baseline, each possible connective is encoded as a binary feature, together with its PoS. Unsurprisingly, the use of only this feature results in a better accuracy for the 'Explicit' classifier (0.54 +/- 0.01) than the 'Implicit' classifier (0.50 +/- 0.04). As noted earlier, implicit sense relations can occur along with explicit conjunctions, when these relations are taken to be inferred from the arguments (and possibly their context), rather than being linked to the explicit conjunction. This property explains why the performance of the 'Implicit' classifier is not much worse: while the connective is not signaling the sense explicitly, the classifier can learn that some implicit senses co-occur with certain explicit connectives/senses. Since discourse adverbials such as 'instead' or 'moreover' can explicitly signal discourse relations, they are also added to the feature set, resulting in a slight increase of accuracy and f-measure for the 'Explicit' classifier (0.56 +/- 0.03 and 0.51 +/- 0.04).

Using PoS tags from the PTB corpus, unigram, bigram and trigram PoS features are implemented. The use of ngrams with $n > 1$

is meant to serve as a proxy for syntactic patterns. The PoS features are also weighted using tfIdf. A single ngram functions as the *term*, an instance of the two arguments of a conjunction represents the *document* (we count how many times a certain ngram occurs in the arguments) and the *inverse document frequency* is calculated using all the training instances. Other properties encoded as features include whether or not a comparative or superlative adjective is present in either arguments and whether there is a modal verb. Negation could serve as a useful feature to identify EXPANSION.DISJUNCTION or COMPARISON.CONTRAST (see example (3)).

(3) *...is now willing to pay higher bank fees and interest,* (ARG1) **but** *isn't likely to boost its $965 million equity contribution* (ARG2). [wsj_2172]

A negation feature has been implemented in its simplest form, checking for the 'un-' affix and for certain predefined negation terms such as 'not'. The negation features also specify in which argument the feature was found.

4.3.2 Results

The connective/adverb features are included in all of the experiments. Table 4 displays all of the results. While the syntactic features only increase the recall of the 'Explicit' classifier, the performance of the 'Implicit' classifier is considerably improved when using the PoS tags of the arguments. The contribution of negation can be seen by comparing rows 7 and 8 in Table 3. For explicit relations, negation improves recall while maintaining precision, while for implicit relations, negation decreases recall while improving precision. The improvement comes from a better detection of the sense EXPANSION.LEVEL-OF-DETAIL.ARG2-AS-DETAIL.

For PoS-trigrams, the Recursive Feature Elimination algorithm shows that for both the 'Implicit' and 'Explicit' classifiers, the twenty highest ranked trigram features all include a CC (coordinating conjunction). This is not surprising because (as noted in Section 3) when an explicitly-conjoined VP has additional inferred senses, the convention is to include the conjunction as part of *Arg2*. The most prominent patterns are either CC followed by either CD (cardinal number) or DT (determiner). The Explicit classifier also includes among its highest ranked PoS-trigrams, three that start with CC and IN (preposition or subordinating conjunction) as in Ex. (4), which reflects a deviation from the typical syntax of conjoined VPs, in which a verb follows the conjunction. This standard pattern appears 1015 times in the corpus.

(4) ... fees they can charge *have plunged to almost nothing* (ARG1) **and** *in some cases are just that* (ARG2). [wsj_1600]

It is interesting to see which of the senses are more easily detected with the inclusion of syntactic features. The 'Implicit' classifier, with its most useful feature-setting of 'trigram PoS-tags', improves on all senses except EXP.SUBST.ARG2-AS-SUBST.. The CC, IN construct mentioned earlier appears mainly in implicit CONTINGENCY.CAUSE.RESULT conjunctions. This sense is also the sense whose f-measure improves the most with the inclusion of syntactic features, from 0.58 to 0.70. The cardinal number feature improves the classification of TEMPORAL.ASYNCHRONOUS.PRECEDENCE relations, where the event specified in Arg1 that precedes that specified in Arg2. A total of 84 implicit tokens contain a cardinal number, many of which describe the movement of stock prices over time. (This is a likely consequence of the content of the WSJ corpus.) An example where the explicit sense is EXPANSION.CONJUNCTION and the implicit sense is TEMPORAL.ASYNCHRONOUS.PRECEDENCE, is:

(5) ... Delta *issued 2.5 million shares of common stock to Swissair* **and** *repurchased 1.1 million shares for use in a company employee stock ownership plan.* [wsj_1011]

The 'Explicit' classifier improves only in recall with the addition of syntactic features. Because the tfIdf weighted unigrams of words and PoS work slightly better than the PoS trigrams, one could conclude that single words or PoS are as much indicative of the sense as syntactic combinations of PoS. A reason for this could be that there is not much syntactic variability in the way a VP constituent can be constructed. COMPARISON.CONTRAST and CONTINGENCY.PURPOSE.ARG2-AS-GOAL get recognized, whereas before they were not, but the f-measure of other senses sinks. There is therefore a trade-off between the classification of more senses and the precision of the individual senses

	Explicit				Implicit			
	Accuracy	Prec.	Rec.	f-measure	Accuracy	Prec,	Rec.	f-measure
2g PoS	0.74 (0.09)	0.74	0.72	0.69 (0.10)	0.60 (0.03)	0.60	0.56	0.55 (0.03)
3g PoS	0.74 (0.07)	0.74	0.72	0.69 (0.09)	0.60 (0.07)	0.60	0.59	0.56 (0.06)
1g words+PoS	0.75 (0.07)	0.75	0.73	0.71 (0.08)	0.60 (0.02)	0.60	0.58	0.55 (0.02)
2g words+PoS	0.73 (0.06)	0.73	0.72	0.68 (0.07)	0.60 (0.03)	0.60	0.59	0.55 (0.04)
2g words	0.73 (0.06)	0.73	0.73	0.67 (0.07)	0.59 (0.09)	0.59	0.56	0.54 (0.08)
3g PoS + 1g words	0.75 (0.09)	0.75	0.73	0.70 (0.10)	0.57 (0.03)	0.57	0.55	0.53 (0.02)
synt. feat. no neg.	0.74 (0.09)	0.74	0.67	0.67 (0.11)	0.45 (0.02)	0.45	0.55	0.37 (0.04)
synt. feat. + neg.	0.75 (0.10)	0.75	0.70	0.68 (0.12)	0.51 (0.02)	0.51	0.50	0.45 (0.03)
3g PoS + synt. feat.	0.74 (0.10)	0.72	0.68	0.68 (0.10)	0.54 (0.04)	0.54	0.52	0.51 (0.04)
Conn/Adv	0.75 (0.07)	0.75	0.68	0.67 (0.09)	0.46 (0.02)	0.46	0.52	0.39 (0.03)

Table 4: Comparison of performance of syntactic features. The number in parenthesis is the confidence interval of the cross-validation score. ('1g' stands for unigram, '2g' for bigram etc., 'synt. feat.' stands for comparative/superlative adjectives and modal verbs)

classified, when using syntactic features for the 'Explicit' classifier.

4.4 Semantic Features

4.4.1 Experiments

In order to exploit the semantic content of the conjunctions, multiple semantic resources are used. These resources generally are semantic representation techniques that are able to reduce the dimensionality of the data. Since the task consists of classifying sense relations between two arguments, a representation of the semantic combination of the two arguments might be suitable. For this purpose the Cartesian product between the corresponding representation of the words in *Arg1* and in *Arg2* is constructed.

VerbNet (Schuler, 2005) features are implemented as the Cartesian product of the verbs in the VPs and also as a tfIdf weighted bag-of-words representation. Since we are working with VP conjunctions the role of the verbs is assumed to be important for the sense of the relation.

BrownCluster classes represent words as semantic clusters, through a hierarchical clustering approach using mutual information (Turian et al., 2010). For the BC features the Brown Clusters from the CoNNL-2016 Shared Task[2], containing 100 clusters, are used. Previous research on discourse relations showed that Brown Clusters are especially useful for the classification of implicit relations (Rutherford and Xue, 2014). BC pairs with a hyponym-meronym relation have been shown to be predictive for the EXPANSION sense (Rutherford and Xue, 2014). Again both the Cartesian product and the bag-of-word representation are implemented.

We used WordNet (Miller, 1992) to analyze the semantic relations and similarity of the words between the two arguments. For this purpose the antonymy, synonymy and hypernymy annotations of WordNet are considered. Every noun and verb in the feature scope is assigned to its disambiguated synset, using Banerjee and Pedersen (2002)'s approach of applying the Lesk algorithm to WordNet. The relational features, such as antonymy, are represented as categorical features containing the respective synset. Similarity between the arguments is encoded into a feature by calculating the normalized shortest-path scores between all the synsets of the two arguments.

4.4.2 Results

The three semantic feature-types, BrownCluster, VerbNet and WordNet, are evaluated in combination with the connectives/discourse adverbials features. Table 5 shows that the 'Implicit' classifier profits the most from the semantic features. This indicates that the semantic information contained in a connective, can, to some extent, be found in in the arguments of implicit relations. For explicit relations, the sense of the relation might not have to be expressed semantically in the arguments. In terms of semantic resources, the TfIdf weighted BC features result in the biggest accuracy and f-measure for the 'Implicit' classifier. The 'Ex-

[2] http://www.cs.brandeis.edu/~clp/conll16st/dataset.html

plicit' classifier shows a minimal improvement in f-measure when adding semantic features. The WordNet features seem to be the least indicative for the 'Implicit' classifier, but still offer an improvement compared to the basic feature set.

The Recursive Feature Elimination shows that most of the highly ranked VerbNet classes contain one or more classes that semantically indicate a verbum dicendi, such as 'approve', 'manner_speaking' or 'indicate'. These verbs seem very indicative of the COMP.CONCESSION.ARG2-AS-DENIER sense, as the denying tends to be expressed in the form of reported speech. The highest ranked BC classes are not as easily analyzed, since the clusters do not have names. Nevertheless, clusters with distinct properties can be identified. One highly ranked cluster contains a lot of hyphen separated adjectives, such as 'double-masted', 'ski-masked' and 'well-built'. Most of the instances in the corpus containing such adjectives display one of the EXPANSION senses, where the adjectives are found in ARG2. Another, more semantically motivated cluster, contains company names such as 'Rossignol' and 'Icelandair', which is probably influenced by the financial domain of the corpus.

4.5 Internal and External Features

In the following, features derived from the arguments and connective are considered *internal features*, while features obtained from outside their scope are considered *external features*. The motivation behind this feature scope exploration comes from the distinction between the senses COMPARISON.CONCESSION and COMPARISON.CONTRAST. While both involve a comparison between *Arg1* and *Arg2*, COMPARISON.CONCESSION is used when one expresses an expected situation which is refuted by the other (either ARG1-AS-DENIER or ARG2-AS-DENIER). The implication of an expectation of a situation might require more textual context or even world-knowledge. Both senses exhibit a similar distribution of connectives (*but* and *implicit connective*), making their distinction even harder. To test whether the internal feature scope is enough or whether some external features could contribute to a better sense classification, a combination of syntactic and semantic features is used on the internal, external and combined feature-scope. The results in Table 6 indicate that the arguments contain all of the information needed to classify the sense of conjoined VPs. Adding the external features on top of the internal features results in about the same performance for the 'Explicit' classifier and in a worse performance for the 'Implicit' classifier. The external features seem to mainly add noise to the feature space. The external scope alone results in the worst 'Explicit' classifier performance until now and stays about the same as the connective/adverb features performance of the 'Implicit' classifier. This experiment therefore showed that for the classification of conjoined VPs the most relevant information is contained in the arguments. At the same time, the assumption that features from the external feature scope are useful to distinguish COMPARISON.CONCESSION and COMPARISON.CONTRAST, has been confirmed. Their classification performance is better when using only external features than when using only internal features (see Table 7). This property could, in future work, be used when a separate classifier is built for every sense.

5 Comparison with off-the-shelf parses

The comparison of feature scope goes hand in hand with the comparison of the classifier's performance on gold-standard data versus automatic parses. While the experiments above have used argument spans provided in the annotated corpus, any practical system will have to rely on whatever conjoined VPs have been identified by its parser. When given a sentence containing a conjoined VP, a parser should produce a parse that includes a VP parent, with VP siblings and a connective or comma in between. While the Stanford Shift-reduce Constituency Parser[3] fulfills this condition, it failed to produce a conjoined VP analysis for 1369 of the 4633 tokens in the corpus. Where it did produce an analysis, the analysis often differed from that in the conjoined VP corpus because of the annotation guidelines. For example, the guidelines indicate that parenthetical and non-restrictive relative clauses (as in Ex. (6)) can be omitted if they don't contribute to the sense relation(s) that hold between the conjuncts (Webber et al., 2016). Reported speech and attribution relations also belong to this category.

(6) It is also *pulling 20 people out of Puerto Rico*, who were helping Hurricane Hugo victims, **and** *sending*

[3] http://nlp.stanford.edu/software/ srparser.shtml

	Explicit				Implicit			
	Accuracy	Prec.	Recall	F-m.	Accuracy	Precision	Recall	F-m.
VN TfIdf	0.72 (0.08)	0.72	0.71	0.69 (0.08)	0.51 (0.02)	0.51	0.47	0.48 (0.02)
VN c.p.	0.73 (0.08)	0.73	0.71	0.69 (0.09)	0.53 (0.05)	0.53	0.51	0.49 (0.06)
BC TfIdf	0.72 (0.09)	0.72	0.71	0.69 (0.10)	0.60 (0.06)	0.60	0.58	0.58 (0.05)
BC c.p.	0.73 (0.06)	0.73	0.71	0.69 (0.07)	0.52 (0.06)	0.52	0.51	0.49 (0.08)
WN	0.74 (0.08)	0.74	0.68	0.67 (0.10)	0.48 (0.02)	0.48	0.45	0.44 (0.02)
Conn/Adv	0.75 (0.07)	0.75	0.68	0.67 (0.09)	0.46 (0.02)	0.46	0.52	0.39 (0.03)

Table 5: Comparison of performance of semantic features (BC = BrownCluster, VN = VerbNet., WN = WordNet, c.p. = Cartesian Product, TfIdf = weighted with TfIdf). For comparison the performance using the basic Conn/Adv features is added.

	Explicit				Implicit			
	Accuracy	Prec.	Recall	F-m.	Accuracy	Precision	Recall	F-m.
Internal	0.73 (0.08)	0.73	0.70	0.69 (0.07)	0. 58 (0.02)	0.58	0.56	0.55 (0.02)
External	0.71 (0.07)	0.71	0.67	0.67 (0.08)	0.47 (0.05)	0.47	0.44	0.44 (0.05)
Int. + Ext.	0.73 (0.08)	0.73	0.70	0.70 (0.10)	0.56 (0.04)	0.56	0.53	0.53 (0.04)

Table 6: Comparison of the two classifiers' performance on features from the internal, external and combined feature scope. For comparison the performance using the basic Conn/Adv features is added.

	Ext.	Int.	Int.+Ext.
Comp.Concess.	0.82	0.81	0.79
Comp.Contrast	0.10	0.04	0.19

Table 7: F-m. for COMPARISON.CONCESSION and COMPARISON.CONTRAST given different feature scopes (using the 'Explicit' classifier).

them to San Francisco instead. [wsj_1899]

Another guideline is that the arguments should follow a parallel structure, where words whose scope encompasses both arguments are not included. This most commonly affects adverbs located in front of *Arg1*.

We carried out two experiments with the annotated VPs and the automated parses – the first simply testing on automated parses and the other, both training and testing on the automated parses. The results from Table 8 show that the performance of a classifier decreases in both experiments. The changes in span and the inclusion/exclusion of adverbs has the biggest effect on recall. This emphasizes the importance of the argument spans for sense classification. The worse performance of the training and testing on the parsed data can also be attributed to the smaller amount of training data available.

Train/Test	precision	recall	f1-m.
goldst/goldst	0.62	0.65	0.60
goldst/parses	0.53	0.43	0.46
parses/parses	0.44	0.45	0.43

Table 8: Results of the goldstandard and automatic parses experiments. Only the tokens containing a conjoined VP analysis in the automatic parses were used for these experiments.

6 Discussion of the full system

In this section the whole two-classifier system, with negative training examples, is evaluated and discussed. The 'Explicit' classifier's performance using the connective/adverb as features could only minimally be improved using tfIdf weighted unigram features of both PoS and words. For the final system this classifier uses only these features. The 'Implicit' classifier uses the tfIdf weighted PoS trigrams and the tfIdf weighted Brown Cluster classes. The full system achieves a precision of 0.66, a recall of 0.64 and an f-measure of 0.59. The customized featureset strategy might not be necessary, as using the same featureset for both classifiers also results in an f-measure of 0.61. To motivate the use of the two-classifier system, we compared it to the performance of a One-Vs-

Rest classifier approach (Pedregosa et al., 2011), where a separate SVM classifier is trained for each sense. The O.Vs.R strategy achieves a precision of 0.74, a recall of 0.52 and f-measure of 0.57. While the precision is higher, the recall and (sense-)weighted f-measure is lower. The advantage of the One-Vs-Rest classifier strategy is a higher accuracy of correctly classified multi-label instances (0.42), whereas the system only classifies 30%. The system is better at classifying individual explicit/implicit senses rather than finding multi-sense combinations. Adding negative instances to the classifiers in order to make them predict whether or not an implicit or explicit sense holds is effective. Many of the correctly predicted senses arise from single-label conjunctions, e.g. the system manages to correctly make the classifiers say when either no explicit or no implicit relation holds. The performance of the system is better than the predefined baseline in Table 2. The f-measure increases from 0.41 to 0.59. The baselines of the individual classifiers, e.g. the 'Implicit' and 'Explicit' classifier, have also been beat. The 'Explicit' classifier, with an accuracy of 0.75 and an f-measure of 0.71 is much better than the baseline of 0.49 and 0.42. The 'Implicit' classifier's baseline improves the most, from an accuracy of 0.37 to 0.6 and an f-measure of 0.2 to 0.56. This is not surprising as we only chose one majority class for all of the implicit instances.

7 Conclusion and Future Work

This paper presents the first work on automatic sense-classification of conjoined VPs and hopefully inspires more research on this topic, further improving the classification performance. Since sense labelling is only a subtask of shallow discourse parsing, future work could be concerned with the construction of a complete discourse parser for conjoined VPs. An improved argument detection system could allow a better characterization of the extent to which errors in argument span make a difference in sense classification.

References

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145. Springer.

Jessica Ficler and Yoav Goldberg. 2016. Improved parsing for argument-clusters coordination. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 72.

Hugo Hernault, Helmut Prendinger, David A DuVerle, Mitsuru Ishizuka, and Tim Paek. 2010. Hilda: a discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33.

Shafiq Joty, Giuseppe Carenini, and T. Raymond Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics, Volume 41, Issue 3 - September 2015*, pages 385–435.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156. Association for Computational Linguistics.

Thomas Meyer, Najeh Hajlaoui, and Andrei Popescu-Belis. 2015. Disambiguating discourse connectives for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(7):1184–1197.

George Miller. 1992. Wordnet: a lexical database for english. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16. Association for Computational Linguistics.

Rashmi Prasad and Aravind Joshi. 2008. A discourse-based approach to generating why-questions from texts. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*, pages 1–3.

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. 2014. Reflections on the penn discourse treebank, comparable corpora, and complementary annotation. *Volume 40, Issue 4 - December 2014*, pages 921–950.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational*

Linguistics, pages 645–654. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon.

Manfred Stede. 2011. Discourse processing. *Synthesis Lectures on Human Language Technologies*, 4(3):1–165.

Rajen Subba and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574. Association for Computational Linguistics.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24. Association for Computational Linguistics.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2016. A discourse-annotated corpus of conjoined vps. *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 22–31.

Deception Detection in Russian Texts

Olga Litvinova and **Tatiana Litvinova**
Voronezh State Pedagogical University
Lenina St, 86, Voronez, Voronezhskaya oblast', 394024
olga_litvinova_teacher@mail.ru
centr_rus_yaz@mail.ru

Pavel Seredin
Voronezh State University
Universitetskaya pl., 1, Voronez,
Voronezhskaya oblast', 394036
paul@phys.vsu.ru

John Lyell
Higher School of Economics
Myasnitskaya ul., 20, Moskva, 101000
jjlyell@gmail.com

Abstract

Psychology studies show that people detect deception no more accurately than by chance, and it is therefore important to develop tools to enable the detection of deception. The problem of deception detection has been studied for a significant amount of time, however in the last 10-15 years we have seen methods of computational linguistics being employed with greater frequency. Texts are processed using different NLP tools and then classified as deceptive/truthful using modern machine learning methods. While most of this research has been performed for the English language, Slavic languages have never been the focus of detection deception studies. This paper deals with deception detection in Russian narratives related to the theme "How I Spent Yesterday". It employs a specially designed corpus of truthful and deceptive texts on the same topic from each respondent, such that N = 113. The texts were processed using Linguistic Inquiry and Word Count software that is used in most studies of text-based deception detection. The average amount of parameters, a majority of which were related to Part-of-Speech, lexical-semantic group, and other frequencies. Using standard statistical analysis, statistically significant differences between false and truthful Russian texts was uncovered. On the basis of the chosen parameters our classifier reached an accuracy of 68.3%. The accuracy of the model was found to depend on the author's gender.

1 Introduction

Deception is defined as the intentional falsification of truth made to cause a false impression or lead to a false conclusion (Burgoon and Buller, 1994). Psychology studies show that all types of people students, psychologists, judges, law enforcement personnel detect deception no more accurate than chance (Bond and DePaulo, 2006). Vrij (2010) pointed out that machines are far outperform humans at detecting deception. Therefore, creation of new automatic techniques to detect deception are vital.

Scientists have been studying deception for a long time, attempting to design text analysis techniques to identify deceptive information. However, it is only very recently that methods of modern computational linguistics and data analysis have been employed in addressing this issue (Newman et al., 2003). With the growing number of Internet communications it is increasingly important to identify deceptive information in short written texts. This poses a great deal of challenge as there are no non-verbal cues in textual information, unlike in face-to-face communication.

Obviously there is no single linguistic feature which can with high accuracy partition deceptive from truthful texts. It is thus important to utilize a combination of certain frequency-based text parameters, making up what can be called a linguistic deception profile. The use of a selection of various parameters is vital in analyzing texts for deceptive information and Vrij was right to say that, a verbal cue uniquely related to deception, akin to Pinocchios growing nose, does not exist. However, some verbal cues can be viewed as weak di-

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 43–52, Valencia, Spain, April 3-7 2017.

agnostic indicators of deceit (2010). In this way, it seems clear that a combination of features is more effective than isolated categories.

To discern deceptive patterns in communication in the field of Natural Language Processing (NLP), over the last 10-15 years, new approaches to deception detection have arisen, relying essentially on the analysis of stylistic features, mostly automatically collected, as with a vast majority of similarly related NLP tasks, for example, in native language identification (NLI), the task of detecting an authors native language form their second language writing (Shervin and Dras, 2015).

Many recent studies involving automated linguistic cue analysis, including studies concerning deception detection, have leveraged a general-purpose, psycho-social dictionary such as Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2007).

Most papers dealing with automated deception detection were performed using English texts with the evaluation of reliability/truthfulness of the narrative being addressed as a text classification task employing machine learning methods. However, more recently in NLP tasks, methods and models are tested across different languages (see Shervin & Dras (2015) an example of such work in the field of NLI).

To the best of our knowledge, the problem of deception detection as an NLP task has not to date been addressed for Russian, which is connected in large part to the lack of applicable data sets. The lack of standard data sets for this task motivated us to construct our own data set a corpus of truthful and deceptive narratives, written in Russian, on an identical topic from the same author. The corpus contains detailed information about each author (gender, age, psychological testing results etc.), and represents an additional contribution to this work. The corpus is now available on request, but in the near future it will be available only on a specially created site.

Using the previously mentioned corpus, a statistically significant difference between truthful and deceptive texts from the same author, written using an identical theme, was discovered. Utilizing these parameters we offer a new approach to the evaluation of the reliability/truthfulness of the Russian written narrative. The classifier was test separately for both men and women.

2 Related Work

Deception detection (in the framework of computational linguistics) is usually conceived of as a text classification problem where a system should classify an unseen document as either truthful or deceptive. Such a system is first trained on known instances of deception. One of the first studies to employ this approach was the one by Newman et al. (2003), who showed that by using supervised machine learning methods and quantitative text parameters as features one can automatically classify texts as deceptive or truthful. The authors obtained a correct classification of liars and truth-tellers at a rate of 67% when the topic was constant and a rate of 61% overall.

Frequently used features have been token unigrams and LIWC lexicon words starting originally with the above paper by Newman. LIWC (Pennebaker et al., 2007) is a text analysis program that counts words in psychologically meaningful categories. LIWC processes text based on 4 main dimensions: standard linguistic dimensions (1), psychosocial processes (2), relativity (3) and personal concerns (4). Within each dimension, a number of variables are presented, for example, the psychosocial processes dimension contains variable sets representing affective and emotional processes, cognitive processes and so forth. Using the LIWC 2015, up to 88 output variables can be computed for each text, including 19 standard linguistic dimensions (e.g., word count, percentage of pronouns, articles), 25 word categories tapping psychological constructs (e.g., affect, cognition), 10 dimensions related to relativity (time, space, motion), 19 personal concern categories (e.g., work, home, leisure activities), 3 miscellaneous dimensions (e.g., swear words, nonfluencies, fillers), and 12 dimensions concerning punctuation information. The default dictionary contains 2300 words, which are used as the basis for output categories. With a few exceptions, the output variables represent only a percentage of the total words found in LIWC dictionary (Pennebaker et al., 2007).

Several studies have relied on the LIWC lexicon to build deception models using machine learning approaches and showed that the use of semantic information is helpful for the automatic identification of deceit. For example, Mihalcea & Strapparava (2009) used LIWC, measuring several language dimensions on a corpus of 100 false and true

opinions on three controversial topics similar to Newman et al. (2003). They achieved an average classification performance of 70%, which is significantly higher than the 50% baseline. It is worth noting that they also tested the portability of the classifiers across topics, using two topics as training sets and the third topic as a test set. The fact that the average accuracy was significantly higher than the 50% baseline indicates that the learning process relies on clues specific to truth/deception, and it is not bound to a particular topic.

In a similar study of Spanish texts (Almela et al., 2013), the discriminatory power of almost all LIWC variables under the first two dimensions (linguistic and psychological processes), the most relevant ones, have been checked (73.6%).

Until now, very little attention has been paid to address the identification of deception based on demographic data using computational approaches because of the scarcity of resources for this task (Prez-Rosas and Mihalcea, 2014). We are aware of only two other resources for deception detection where demographic data is available (Prez-Rosas and Mihalcea, 2014; Verhoeven and Daelemans, 2014).

In the study (Fornaciari et al., 2013) the authors combined deception detection and personality recognition techniques, in order to get some insight regarding the possible relation between deception and personality traits from the point of view of their linguistic expression. They found that the machine learning models perform better with subjects showing certain kinds of personality traits (when taking into account the author's communication style, deceptive statements are more easily distinguished). However, as the authors themselves suggest, the relatively small amount of respondents allowed them to obtain only a few types of personalities.

In the study by Levitan et al. (2016) for oral speech it was shown that for deception detection, when they included binned NEO-scores, as well as gender and language, in addition to the prosodic and LIWC feature sets, accuracy of the classifier went up to 65%, i.e. there is a 25% relative increase over the majority class baseline and a 13% absolute increase.

For this particular study, we have made use of Linguistic Inquiry and Word Count. We used a LIWC Russian dictionary and also designed our own dictionaries (see explanations below). The analysis was performed along 104 parameters used to distinguish truthful and deceptive texts.

3 Data and Settings

Firstly, in order to address the subject at hand, we must study a corpus containing truthful and deceptive texts. Collecting this type of text corpus constitutes a scientific task in itself (Fitzpatrick and Bachenko, 2012). Most text corpora being studied presently have a volume of limitations caused by too few respondents, as well as a paucity of deceptive and truthful texts written by the same individual and for the data set as a whole, due in large part to the difficulty of obtaining a control sample of texts in which the same author tells the truth for the sake of comparison. What is important in developing methods of lie detection in texts is to identify changes in the idiolect of the same individual when they produce both deceptive and truthful texts on the same topic. Additionally, as was noted, most corpora contain only English texts.

Another downside of the existing corpora is the shortage of detailed metadata providing the authors personal information (gender, age, education level, psychological testing data, etc.) to establish the effects of personality traits on how deceptive texts are produced.

In our paper we have used a text corpus Russian Deception Bank. It was launched in 2014 as part of a text corpus called RusPersonality (Litvinova et al., 2016). Deception Bank currently contains truthful and deceptive narratives (average text length is 221 words, SD = 15.2) of the same individuals on the same topic (How I spent yesterday) (see example in Table 1).

Since it was not a spontaneously produced language, it was deemed necessary to minimize the effect of the observers paradox by not explaining the ultimate aim of the research to the participants. In addition, to motivate them, the respondents were told that their texts (without information of which of them were truthful and which were not) would be evaluated by a trained psychologist who would attempt to tell a truthful text from a deceptive one. Each respondent whose texts would not be correctly evaluated would be awarded with a cinema ticket voucher.

The number of the authors is N = 113 as of now (46 males, 67 females, university students, all native speakers of Russian) and there are plans

Truthful Text	Deceptive Text
So here we were in Piter and went to the apartment that we had booked, it was not far from the city centre. Having dropped off our stuff, we went on a walk around the city centre and grabbed something to eat. Well, actually every afternoon we spent here was pretty much the same. In the evening we would go to any Pub or Bar and killed time there. Yes, killed time because it was not much fun. Maybe its because the people around werent much fun. Of course it was interesting to visit the museums and other sights of the city but I cant say that really left an impression that it was supposed to and all in all, I didnt feel too happy throughout that trip.	Having come to Piter, first thing we went to the apartment that we had booked, it was in the city centre, straight in Nevskiy, our window overlooked the beautiful views of Piter, especially in the evening when the sun went down, it was very beautiful. Of course you can spend ages walking the streets of the city and never get tired, while you are walking, you cant help being happy about everything you see around you. Every evening we would drive around different places in the city and sure thing, we dont have any clubs or pubs like that back home and I dont think we ever will. The way this city makes you feel is just special.

Table 1: Sample statements from the same author

to extend it. Apart from truthful and deceptive texts by each individual, Russian Deception Bank (as well as all the texts in RusPersonality) comes with metadata which provides detailed information about their authors (gender, age, psychological testing results). Hence, the annotated Russian Deception Bank will enable authors personal features (psychological and physical) to be considered as a factor contributing to the production of their deceptive texts.

We argue that these data are critical in designing an objective method of identifying intentionally deceptive information (Levitan et al., 2016). Each text was entered into a separate text file, and misspellings were corrected. Each of the 226 text files was analyzed using LIWC 2015 and a Russian language dictionary based on LIWC2007.

We have employed a basic Russian language dictionary that comes with the LIWC software and additionally developed our own users dictionaries (see explanations below). It's worth noting that the program's Russian dictionary is a simple translation of the corresponding English LIWC dictionary. For our study we selected categories that were the least dependent on the content of the texts. Hence the following parameters were selected:

- I STANDARD LINGUISTIC DIMENSIONS (19),
- II PSYCHOLOGICAL PROCESS DIMENSIONS (Affective Processes - 5, Cognitive Processes - 8, Perceptual Processes - 3, Relativity - 3),
- All Punctuation parameters (11).

Users dictionaries were also compiled according to the user manual:

- a dictionary of 20 most frequent function words in Russian Freq FW (20 parameters account for the uses of each word in a text and 1 parameter represents the proportion of the total uses of all such words in a text)
- a dictionary of demonstrative pronouns and adverbs Deictic (1 parameter accounts for the proportion of these words per to the total word length of a text)
- discourse markers DM (10)
- a dictionary of intensifiers and downtowners Intens (2 parameters)
- a dictionary of pronouns as parts of speech Pron (10)
- a dictionary of perception vocabulary PerceptLex (1 parameter)
- a dictionary of pronouns and adverbs describing the speaker Ego (I, my, in my opinion) (1 parameter)
- a dictionary of emotional words Emo (negative and positive, 2 parameters).

All in all, there are 104 parameters. The users dictionaries were compiled using the available dictionaries and Russian thesauri.

It was necessary to compile these particular dictionaries owing to the fact that the Russian dictionary that came with the software was a translation of a corresponding English dictionary and did not stand independent testing, i.e. if all the variables from the first group are identified unambiguously, there are doubts as to the semantic category of the second group and thus they have to be evaluated independently and objectively. The results were processed using SPSS 23.0 software.

4 Experiments

Originally we excluded the parameters that had a frequency of less than 50%. Here frequency of a parameter is defined as a ratio of non-zero values of a parameter to the number of all of the analyzed texts (both truthful and deceptive ones). The selected parameters are identified in the Table.

Further on we calculated and evaluated the variation coefficient of the text parameters that indicates the range of a linguistic parameter in the texts by the same author (Viktor V. Levitsky , 2004). This can be done using the following ratio:

$$V = \frac{\sum_i^n * \frac{|x_{Ti} - x_{Di}|}{x_{Ti}}}{n} * 100\% \qquad (1)$$

where x_{Ti} is the value of the i-th parameter in a truthful text, x_{Di} is the value of the i-th parameter in a deceptive text, and n is a selection size. The computed variation coefficients are shown in Table 2.

A statistical analysis (see Table 2) showed that the computed variation coefficient for the selected parameters ranges significantly. The parameters with correlation coefficient over 50 % were excluded at the next stage (see Levitsky (2004); Litvinova, (2015)).

In order to understand how the parameters of truthful and deceptive texts by the same author change in relation to the absolute value, we calculated the averaged values of each parameter. Table 2 presents a relative change in each parameter in deceptive texts in relation to truthful ones (in percentages).

In order to determine which of the originally selected text parameters could be used in further calculations, we tried to establish a connection between the variation coefficients of the text parameters, frequencies of the parameters in the texts as well as the difference between the average values of the text parameters in a selection of truthful and deceptive texts. Using the methods of correlation analysis, we found that for a statistical significance level p¡0.05 there is no connection between the frequency of a parameter and a difference between the average values of truthful and deceptive texts. At the same time the calculation of the Pearson correlation coefficient for frequencies of the text parameters and their variation coefficient showed that there is a considerably strong connection r¿0.9 at p¡0.05 (a linear dependence between the two values). Therefore there is one important conclusion to be made: the use of only average values of text parameters in a selection is not always the best option as it does not allow for the distribution of a certain parameter in deceptive and truthful texts by the same author. In order to consider a type of the distribution of text parameters in the corpora of truthful and deceptive texts we used one of the most effective criteria for checking the normality is the use of the Shapiro-Wilk test for normality as it is stronger compared to the alternative criteria for small samples. However, some of the text parameters in deceptive texts (Sixltr, AllPunc, PersPronUser) change their distribution differently. Only the parameters with the following characteristics were chosen for the model to evaluate truthful and deceptive texts:

they are frequent (i.e. occur in no less than half of the texts); vary reasonably in the texts by the same author (on average in a selection); have normal distribution (since we have Student's statistics as a basis of our classifier).

It should be noted that in order to design the models, the parameters that are normally distributed in the corpus of truthful texts were employed. According to the calculation, only 10 parameters are normally distributed in truthful texts (see Table 3).

Hence in deceptive texts in Russian compared to truthful ones on the same topic there are more verbs, conjunctions overall, specifically the conjunctions and, as well as words for cognitive processes overall and inclusive words in particular, additional discursive markers, pronominal nouns, and more personal pronouns (even though it was only revealed at the 10% significance level). In truthful texts there are more prepositions and punctuation marks. Consequently, characteristic features of deceitful texts from a morphological

Parameter	Frequencies in truthful texts, %	Frequencies in deceptive texts, %	Difference in the averaged values of the parameters in deceptive texts in relation to truthful ones, %	Variation Coeff.
Words per sentence (WPS)	100	100	5.53	22.47
Words¿6 letters (Sixltr)	100	100	-0.6	15.09
Function words (FW)	100	100	1.59	11.32
Total pronouns	97	99	2.59	29.14
Total pers pronouns	97	98	6.11	29.78
1st pers singular	94	93	5.71	38.54
1st pers plural	68	67	4.94	86.06
3rd pers singular	88	86	5.39	70.40
3rd pers plural	87	83	-7.57	70.72
Verbs	100	100	3.02	27.37
Adverbs	88	89	-3	59.92
Prepositions	100	100	-1.03	19.70
Conjunctions	100	100	3.94	27.93
Negations	83	76	2.91	87.17
Quantifiers	95	89	1.75	64.971
Numbers	56	52	13.46	121.19
Cognitive Processes	100	100	3.14	21.711
Insight	90	91	7.78	58.24
Causation	87	84	4.54	73.46
Discrepancy	65	60	-7.95	99.84
Tentative	78	76	2.73	83.32
Certainty	86	85	0.5	60.72
Inhibition	52	50	-3.92	117.41
Inclusive	100	98	5.24	33.96
Exclusive	83	76	2.73	72.3
Perceptual Processes	92	88	-11.67	58.07
Seeing	60	58	-13.75	95.76
Hearing	66	70	12.90	92.06
Feeling	54	47	-25.75	117.58
Space	100	99	1.18	25.5

Parameter	Frequencies in truthful texts, %	Frequencies in deceptive texts, %	Difference in the averaged values of the parameters in deceptive texts in relation to truthful ones, %	Variation Coeff.
Time	100	98	-4.45	32.91
All Punctuation (AllPun)	100	100	-6.78	16.21
Period	100	100	-4.83	25.19
Comma	100	99	-1.52	32.66
Dash	79	61	-31.62	86.14
Freq FW	100	100	-0.05	10.83
(and)	98	98	7.31	35.39
(in)	97	96	-1.72	45.3
(not)	82	76	-2.36	85.88
(on)	89	88	4.97	62.78
(with)	80	86	10.71	74.6
(that)	77	71	5.88	75.77
(over)	50	58	16.36	114.15
(but)	68	58	-6.66	101.99
(like)	65	65	1.17	94.18
Deictic	95	90	-5.55	94.18
DM Additions	97	98	7.67	36.16
DM Substitutions	74	76	27.96	77.84
Intensifiers	63	65	-17.77	91.19
Noun-like Pron	98	96	5.65	40.68
Adverb-like Pron	85	85	2.18	67.77
Adjective-like Pron	77	75	11.01	83.49
Number-like Pron	53	52	-21.31	110.55
PersPronUser	96	93	6.91	45.34
PerceptLex	58	50	-17.64	106.98
Ego	92	91	2.8	43.24
Positive Emo	80	75	-1.85	86.88

Table 2: Text analysis data

Parameter	Mean	t	p
Total pers pronouns		1.655	0.1
D	10.932		
T	10.298		
Verbs		3.979	0.0001
D	15.460		
T	15.305		
Prepositions		-3.352	0.001
D	13.366		
T	13.484		
Conjunctions		5.848	0.0001
D	8.496		
T	8.163		
Cognitive Processes		11.916	0.0001
D	18.050		
T	17.488		
Inclusive		9.236	0.0001
D	9.202		
T	8.735		
AllPun		-3.382	0.001
D	20.320		
T	21.801		
(and)		11.726	0.0001
D	3.948		
T	3.685		
DM Additions		12.915	0.0001
D	3.912		
T	3.658		
Noun-like Pron		5.798	0.0001
D	7.610		
T	7.212		

Table 3: Statistical differences between deceptive (D) and truthful (T) texts

standpoint are a greater amount of verbs, personal pronouns, pronominal nouns, conjunctive relationship markers, and a lesser amount of prepositions and punctuation marks. As it seems, this is connected to the fact that texts which contain such characteristics demand less cognitive effort in their creation, however this is merely a proposition and, of course, would need to be verified.

The basis of this model is Rocchio classification. For text classification we first created two centroids $[ST1...ST10]$ and $[SD1...SD10]$, based on the previously attained values of STi and SDi. These serve as averages of all the 10 various chosen parameters in both the truthful and deceitful texts..

For each text, in order to find whether they are or are not truthful, we then need to find the vector S, which consists of all elements Si, i.e. the 10 aforementioned parameters. Our classifier then determines the truthfulness of a text based on the similarity between the vectors of the test documents and the centroids specific to truthful and deceitful texts.

To measure the similarity of the the test set text vectors and the centroids we utilized the cosign similarity of the vector and the centroid. However, our experiment shows that in this case purely measuring cosine similarity actually has a very weak ability to classify texts. Thus with out experiment we decided it was better to use the function listed below (2), which represents a hybrid of both the Euclidean distance formula and the cosine simi-

larity between two vectors.

The similarity of vector S and centroid S_T is measured thus:

$$\chi^2_T = \frac{1}{n}\sum_i^n \frac{(S_{Ti} - S_i)^2}{S_{Ti}} \quad (2)$$

Analogously, the similarity of vector S and centroid S_D is measured as such:

$$\chi^2_D = \frac{1}{n}\sum_i^n \frac{(S_{Di} - S_i)^2}{S_{Di}} \quad (3)$$

We will assume, that in order to determine the type of text, it is sufficient to compare the values of χ^2_T and χ^2_D. The text in question will be classified as deceitful if $\chi^2_T > \chi^2_D$ and classified as truthful if $\chi^2_T < \chi^2_D$

In order to test this approach, before designing the model, the texts were divided into the learning and test sets (70 %, i.e. 158 texts for train and 30 %, i.e. 68 texts for tests). In order to evaluate the suggested model the overall accuracy, which is the percentage of texts that are classified correctly, was computed. The accuracy of the suggested approach evaluated on the total test corpus was 68.3 %. Since data set has an equal distribution between truthful and deceptive texts, the baseline is 50 %.

In our study the accuracy of the model was tested individually for males and females and so was the overall one. The classification accuracy for males was 73.3 % and 63.3 % for females. Hence the analysis indicates that models for detecting deception in written texts could be further improved by considering the characteristics of their authors.

5 Conclusion

The average classification accuracy of 68.3 % although higher than the 50% baseline indicates that classification task is difficult and more research is needed to discover what methodology could be appropriate to improve the results. The analysis revealed that models for detecting deception in written texts could be significantly improved by considering the characteristics of their authors. Males and females lie in different ways. Thus models should be further designed for deceptive/truthful texts by males/females, for peoples of different ages, different psychological profiles in order for them to be more accurate. This is a promising research field, however it has not been properly addressed as part of text-based deception detection because of the scarcity of resources for this task. We assume that the corpus of deceptive and truthful Russian texts with metadata providing various personal information about their authors (gender, age, education, results of psychological and neuropsychological testing and so forth) would contribute to further improvements in this field. Currently we are extending our corpus using real texts - recordings of job candidates in one of Russia's largest industrial companies. All of the candidates took a series of psychological tests. Parts of the interviews were classed as truthful/deceptive using polygraph readings, collection of extra information about the candidates as well as follow-up interviews. To the best of our knowledge, the corpus being designed has no equivalents.

The corpus is to be further expanded by increasing the number of texts as well as respondents. The features of the production of deceptive texts depending on the gender, age and psychological characteristics of their authors are also to be identified. There are plans to design a corpus of deceptive and truthful texts in the first language (Russian) as well as the second language (English) by the same author in order to identify possible structural and lexical differences between the linguistic expression of deceit in both languages.

Further we plan to expand upon our list of parameters and utilize various machine learning algorithms for classifying truthful and deceitful texts, and then compare these results to the method mentioned in this paper.

Acknowledgments

This research is supported by a grant from the Russian Foundation for Basic Research, N 15-34-01221 Lie Detection in a Written Text: A Corpus Study.

References

Angela Almela, Rafael Valencia-Garca, and Pascual Cantos. 2013. Seeing through deception: A computational approach to deceit detection in written communication. *Linguistic Evidence in Security, Law and Intelligence (LESLI)*, 1(1).

Charles F. Bond and Bella M. DePaulo. 2006. Accuracy of deception judgments. *Personality and Social Psychology Review*, 10(3):214–234.

Judee K. Burgoon and David B. Buller. 1994. Interpersonal deception: Ill effects of deceit on perceived communication and non-verbal behavior dynamics. *Journal of Nonverbal Behavior*, 18(2):155–184.

Eileen Fitzpatrick and Joan Bachenko. 2012. Building a data collection for deception research. In Eileen Fitzpatrick, Joan Bachenko, and Tommaso Fornaciari, editors, *Proceedings of the EACL Workshop on Computational Approaches to Deception Detection*, pages 31–38.

Tommaso Fornaciari, Fabio Celli, and Massimo Poesio. 2013. The effect of personality type on deceptive communication style. In *Intelligence and Security Informatics Conference (EISIC)*. IEEE.

Sarah Ita Levitan, Yocheved Levitan, Guozhen An, Michelle Levine, Andrew Rosenberg, Rivka Levitan, and Julia Hirschberg. 2016. Identifying individual differences in gender, ethnicity, and personality from dialogue for deception detection. In *NAACL Workshop on Computational Approaches to Deception Detection*, San Diego.

Tatyana Litvinova, Olga Litvinova, Olga Zagorovskaya, Pavek Seredin, Alexander Sboev, and Olga Romanchenko. 2016. "ruspersonality": A russian corpus for authorship profiling and deception detection. In *International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT 2016)*, pages 1–7, St. Petersburg, RU.

Tatiana Litvinova. 2015. On the problem of stability of parameters of idiostyle. In *Proceedings of Southern Federal University*, pages 98–106.

Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the Association for Computational Linguistics (ACL-IJCNLP 2009)*, pages 1–7, Singapore, SG.

Matthew L. Newman, James W. Pennebaker, Diane S.Berry, and Jane Richards. 2003. Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin*, 29(5):665–675.

James W. Pennebaker, Cindy K. Chung, Molly Ireland, Amy L. Gonzales, and Roger J. Booth, 2007. *The Development and Psychometric Properties of LIWC2007*. Linguistic Inquiry and Word Count (LIWC), Austin, TX.

Vernica Prez-Rosas and Rada Mihalcea. 2014. Gender differences in deceivers writing style. *Journal Lecture Notes in Computer Science*, 8856:163–174.

Malmasi Shervin and Mark Dras. 2015. Multilingual native language identification. *Natural Language Engineering*, 1:1–53.

Ben Verhoeven and Walter Daelemans. 2014. Clips stylometry investigation (csi) corpus: A dutch corpus for the detection of age, gender, personality, sentiment and deception in text. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC14)*, Reykjavik, IS. European Language Resources Association (ELRA).

Viktor V. Levitsky . 2004. *Quantitative methods in linguistics*. Nova Kniga, Vinnytsia, UKR.

Aldert Vrij. 2010. *Detecting lies and deceit: Pitfalls and opportunities*. John Wiley and Sons, Chischester, UK, 2nd edition.

A Computational Model of Human Preferences for Pronoun Resolution

Olga Seminck
LLF (CNRS)
8 Place Paul-Ricoeur, 75013 Paris
Université Paris Diderot, Paris 7
5 rue Thomas Mann, 75013 Paris
olga.seminck@cri-paris.org

Pascal Amsili
LLF (CNRS)
8 Place Paul-Ricoeur, 75013 Paris
Université Paris Diderot, Paris 7
5 rue Thomas Mann, 75013 Paris
pascal.amsili@linguist.univ-paris-diderot.fr

Abstract

We present a cognitive computational model of pronoun resolution that reproduces the human interpretation preferences of the Subject Assignment Strategy and the Parallel Function Strategy. Our model relies on a probabilistic pronoun resolution system trained on corpus data. Factors influencing pronoun resolution are represented as features weighted by their relative importance. The importance the model gives to the preferences is in line with psycholinguistic studies. We demonstrate the cognitive plausibility of the model by running it on experimental items and simulating antecedent choice and reading times of human participants. Our model can be used as a new means to study pronoun resolution, because it captures the interaction of preferences.

1 Introduction

Pronoun resolution has been studied in the frame of theories of formal grammar, corpus studies, experimental psycholinguistic studies and NLP systems.[1] But much of the findings made about the phenomenon are not shared between these disciplines. This paper takes a step towards more interdisciplinarity between the fields of NLP and psycholinguistics by building a cognitive computational model of pronoun resolution. As Keller (2010) argues convincingly, both the domains of NLP and psycholinguistics can benefit from such models. On the one hand, there is a very rich psycholinguistic literature of which researchers in the domain of NLP are often not aware. NLP techniques might improve if this literature is taken into account. On the other hand, cognitive computational models are a new means to perform psycholinguistic research: by implementing different models that represent different theories, a comparison can be made by looking at the behavior of the models on actual experimental human data.

On the topic of pronoun resolution some cognitive computational models have already been proposed. Frank et al. (2007) proposed a model that resolves ambiguous pronouns based on human interpretation biases (preferences) — such as the *first mention bias*[2]— and world knowledge. They used a so-called *micro-world*: a collection of very detailed world knowledge for a small set of events. Their model was able to simulate reading times, but it remains an open question to what extent the model can be scaled up (Frank et al., 2007).

Kehler and Rohde (2013) proposed a probabilistic model to predict human interpretation biases. Their model, based on world knowledge and information structure, predicts the probability that a given referent is mentioned next. They tested the model on human data from completion tasks[3] and showed that the model could accurately predict the human data.

Dubey et al. (2013) developed a model based on surprisal. Surprisal is a measure that is high when infrequent, or unexpected, events happen. According to Surprisal Theory (Hale, 2001), the surprisal of syntactic structures reflects their cognitive processing cost. That is to say that infrequent syntactic structures are more difficult to process for humans than frequent ones. Demberg and Keller (2008) showed that syntactic surprisal is a relevant factor to model reading times on corpus. In the model of Dubey et al. (2013) syntactic surprisal

[1] In the latter domain nowadays mostly in the form of the coreference resolution task, of which proper pronoun resolution is only a part.

[2] A character that is named first in the sentence is the preferred interpretation of ambiguous pronouns.

[3] In a completion task participants have to complete a text of which only the beginning is given.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 53–63, Valencia, Spain, April 3-7 2017.

is enriched by surprisal coming from coreference. Surprisal is higher when a new referent is introduced and lower when an old one is re-mentioned. Dubey et al. (2013) show that their enriched measure of surprisal is better in explaining the variance in reading times recorded on corpus than a standard measure of only syntactic surprisal.

Inspired by Dubey et al. (2013), we aim for a model of pronoun resolution that can run on natural texts and explain reading times. A second aim for our model is that it can account for human preferences discovered in the psycholinguistic literature. Based on these criteria, we build a model inspired by NLP pronoun resolution systems (Soon et al., 2001). The factors of influence on pronoun resolution are represented as weighted features. This provides a way to assess their relative importance and allows to study their interaction.

In this paper we demonstrate our model by running it on items used in psycholinguistic experiments about human preferences. We first show that the strength of human preferences corresponds to the weights our model associates to different factors influencing pronoun resolution. Second, we study how the model chooses antecedents for pronouns and see that it makes choices similar to humans. Finally, we simulate reading times by formulating a metric of processing cost based on our model.

2 Preferences Modeled in This Work

We chose to model two preferences that operate in English in this work: the Subject Assignment Strategy and the Parallel Function Strategy. We made this choice because of the feasibility of the implementation: both preferences rely only on syntactic mechanisms, so no semantic representation needed to be implemented.

The Subject Assignment Strategy states that, if a pronoun is ambiguous (*i.e.* has more than one antecedent candidate compatible in gender and number), it will be resolved to the antecedent candidate that is in the subject position (Crawley et al., 1990). So for both of the following examples the Subject Assignment Strategy predicts that the antecedent of the pronoun is *John*.

(1) a. John hit Fred and [he]$_{resolve}$ kicked Ellen.
b. John hit Fred and Ellen kicked [him]$_{resolve}$.

According to the Parallel Function Strategy, an ambiguous pronoun is resolved to the antecedent candidate that has the same syntactic function (Smyth, 1994). So according to this second strategy, in example (1-a) *he* will be resolved to *John*, whereas in (1-b) *him* will be resolved to *Fred*.

Evidence for both the Subject Assignment Strategy and the Parallel Function Strategy is not new and comes from early studies from the 1970's (Hobbs, 1976; Sheldon, 1974, among others). However, the interaction between both strategies was investigated more recently by Crawley et al. (1990). They performed two experiments with stimuli like the one in (2), where an ambiguous pronoun in the direct or indirect object position had to be resolved to either a character in the subject position (Brenda) or a character in the object position (Harriet). They chose not to study pronouns occupying the subject position, because both the Subject Assignment Strategy and the Parallel Function Strategy make the same predictions for these pronouns. Instead, they studied resolution of ambiguous pronouns in the direct and indirect object function to see the influence of both the Subject Assignment Strategy and the Parallel Function Strategy.

(2) Brenda and Harriet were starring in the local musical. Bill was in it too and none of them were very sure of their lines or the dance steps. Brenda copied Harriet and Bill watched [her]$_{resolve}$.

They found that only the Subject Assignment Strategy was used in pronoun resolution. However, different studies that followed up their paper, such as Smyth (1994) and Stevenson et al. (1995), found strong evidence for the existence of the Parallel Function Strategy alongside the Subject Assignment Strategy. They criticized the fact that many items used by Crawley et al. (1990) weren't exactly parallel: in many items none of the potential antecedents occupied exactly the same syntactic function as the pronoun. For example in item (3) there is no antecedent candidate in the direct object position (*Monica* is in an indirect object position).

(3) Cheryl and Monica were members of the local peace group. Steven had just joined and wasn't very involved yet. Cheryl spoke to Monica about the next meeting and Steven questioned [her]$_{resolve}$ about it.

With new experiments, Smyth (1994) and Stevenson et al. (1995) established the influence of the Parallel Function Strategy. They even suggested that it overrules the Subject Assignment Strategy

if it can be applied.

In our study we build a model of pronoun resolution that can account for some of the findings of Crawley et al. (1990) and of Smyth (1994). More precisely, we run our model on the items of Crawley's experiment and of Smyth's second experiment. [4]

3 Model of Pronoun Resolution

We used a classifier that proceeds according to a probabilistic version of the pair-wise algorithm (Soon et al., 2001). We only account for third person singular personal pronoun resolution in order to approach the psycholinguistic domain where pronoun resolution is most often restricted to these type of pronouns. The third person pronouns can be viewed as different from the first and the second as the latter are deictic rather than anaphorical.

3.1 Resolver

The pairwise resolver is a logistic regression classifier that gives the probability that a pair of a pronoun and an antecedent candidate are coreferent. We chose it for its straightforward interpretation of feature weights, indicating the influence of factors in pronoun resolution. We trained it on examples of pairs of coreferent and non-coreferent mentions. We used the method of Soon et al. (2001) to sample training examples: to get positive training examples (coreferent pairs), each pronoun is coupled to its closest antecedent. To get negative training examples, the pronoun forms a pair with every mention occurring between its closest antecedent and itself.

3.2 Corpus

We trained the resolver on the English newswire part of the Ontonotes 5.0 corpus (Pradhan et al., 2011). This genre approximated the psycholinguistic items the best among the available genres in Ontonotes. A particularity of the corpus is that singleton mentions (referential expressions that are only mentioned once) are not annotated. We resolved this problem by simply considering as a singleton mention every maximal noun phrase that did not overlap with an annotated mention and that was not a pronoun. Moreover, since Ontonotes is not annotated for number nor gender, we had to add (automatically) an annotation for number and gender to the mentions in the corpus. [5]

3.3 Features

The aim of our model is to have interpretable features and not to have the best score on a pronoun resolution task. We proceeded in three steps to establish the features of our classifier. First, we defined a list of standard features for pronoun resolution — inspired by coreference resolution literature (Denis and Baldridge, 2007; Recasens and Hovy, 2009; Soon et al., 2001; Yang et al., 2004) — that we could retrieve in our corpus. [6] It is important to point out that, although we made up our feature list by looking at literature from Natural Language Processing, the features in the list are also discussed in psycholinguistic literature. For example, distance features and part of speech features are discussed in literature about antecedent *saliency* (Ariel, 1991).

Among all the features, we made sure we included the features necessary to test the two preferences investigated in this paper. For the Subject Assignment Strategy, we used a feature that checks whether the antecedent candidate is in the subject position. We implemented the Parallel Function Strategy by a boolean feature of *syntactic path match* that states whether the antecedent candidate and the pronoun have the same path in the syntactic parse tree from the node where the mention is attached to the root of the tree. A simple illustration of this is given in Figure 1 where the syntactic paths of two mentions are given.

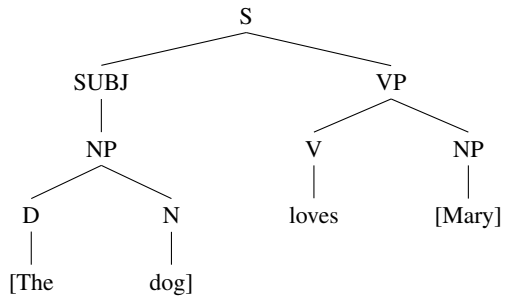


Figure 1: A syntactic tree with two mentions: *the dog* and *Mary*. Syntactic path for *the dog*: [SUBJ, S]. Syntactic path for *Mary*: [VP, S].

[4] We chose these experiments because in the remaining experiments of Smyth (1994), and also in the experiments of Stevenson et al. (1995), a different definition of the Parallel Function Strategy has been used.

[5] The procedure of gender/number annotation we chose is explained in section B of the supplementary materials.

[6] A list of these features can be found in section A of the supplementary materials.

The second step of defining our features consisted in eliminating features too sparsely represented in our training corpus to be adequately learned. As a rule of the thumb we decided to exclude features with a frequency smaller than 0.5%, meaning that every feature should be attested at least 36 times in the training data.

As a last step we checked the significance of our features and removed features that were not significant, because their interpretation is difficult. The model with the features we selected can be found in Table 1.

	Estimate	Signif.
(Intercept)	-2.3533	***
match in gender	2.4206	***
match in number	0.2430	*
m_1 is a subject	1.5142	***
match in syntactic path	1.7318	***
m_1 is a proper noun	0.5007	***
m_1 is a possessive pronoun	1.9037	***
m_1 is a personal pronoun	0.7647	***
words between m_1 and m_2	-0.0114	***
m_1 & m_2 in the same sentence	0.3587	***
length of syntactic path m_1	-0.1361	***
m_1 is determined	-0.2825	*
m_1 is undetermined	-0.4422	**
m_1 has a demonstrative determiner	0.6045	*
m_1 is a common noun	-0.8967	***
m_1 spans m_2	-3.4372	***
length in words of m_1	-0.0201	*
m_1 is a geopolitical entity	-1.2885	***
m_1 is a date	-1.9416	***

Table 1: The selected model of the pronoun resolver. Each factor influencing pronoun resolution has an estimated weight associated that indicates its influence. m_1 refers to the antecedent candidate, m_2 to the pronoun. Significance codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1.

3.4 Evaluation

We divided the corpus into a training set, a development set and a test set. We tested the model's performance on all three of the sets by measuring the accuracy of the identification of antecedents of the third person singular personal pronouns in the corpus. The accuracy and size for each subcorpus can be found in Table 2.

An important question is whether these results are satisfactory. Our results are difficult to compare against state-of-the-art work in coreference resolution, because we concentrate on third person personal singular pronouns only. This means that our system does not form coreference chains and that its performance cannot be measured using standard coreference evaluation metrics, such as MUC, B3, or CEAF (Luo, 2005). A second difference with a more standard approach is that we do not have a module of mention detection. Instead, we use the gold mention annotation and the singleton mentions we extracted (see section 3.2).

Sub-Corpus	Nb. Texts	Nb. Pronouns	Accuracy
Training	476 (60%)	1756	61.15
Development	158 (20%)	558	65.41
Test	158 (20%)	617	61.26

Table 2: The accuracy of the resolver for finding the correct antecedent of the pronoun on the training, development and test set.

This said, we still want to have an indication about the performance of our classifier. The study of Yang et al. (2004) is the most comparable we found to ours, although they used a module for mention detection. Yang et al. (2004) trained different types of systems to perform third person pronoun resolution and reported accuracy, in their paper indicated by the metric of *success*. When they tested on the MUC-6 corpus this metric was between 70.0 and 74.7 for the different systems they developed. When tested on the MUC-7 corpus the metric laid between 53.8 and 62.5. We estimate that, given these numbers, the performance of our model is slightly worse, or comparable.

An error analysis we conducted indicated that most of the errors made by the resolver concerned the pronoun 'it' (about half of the errors). We observed that if we excluded 'it' from resolution the pronoun resolver's accuracy increased by ≈ 16 points. Our error analysis also indicated that a part of the errors comes from our automatic gender annotation: it seems that many coreference chains contain mentions of several genders at once. Nevertheless, we think that the performance on masculine and feminine pronouns of our system is good enough for the purpose of our experiments that include only masculine and feminine pronouns.

3.5 Interpretation of the Model

The weights of the logistic regression model in Table 1 predict the preferences the classifier will show on experimental data. Looking at the feature of syntactic path match and the feature that checks if the first mention is in the subject position, we see that both features have a positive weight; but we can also see that the first is stronger than the second, suggesting that parallel roles are of a greater

impact than the subject position of the antecedent. From this data we can hypothesize that the Subject Assignment Strategy exists alongside the Parallel Function Strategy, and that the Parallel Function Strategy, if applicable, has a stronger influence that can overrule the Subject Assignment Strategy.

4 Antecedent Choice for Pronouns

To test the cognitive plausibility of our model, we ran it on the experimental items of Crawley et al. (1990) and the items of the second experiment of Smyth (1994) and looked if it chose the same antecedents as humans did. That is to say that we compared the model's frequencies of assigning pronouns to subjects and objects with human frequencies.

4.1 Items

For each type of item we give two examples to illustrate the type of experimental items used. Before running the model, we manually annotated the items with coreference and named entity information. For the syntactic annotation we first ran the Stanford Parser (Klein and Manning, 2003) and then corrected the parses manually.

4.1.1 Crawley's Ambiguous Items

From the experiment of Crawley et al. (1990) we have 40 ambiguous items. Ambiguity is produced by gender. The pronoun that has to be resolved is presented in the last sentence in the direct or indirect object position.

1. John and Sammy were playing in the garden. One of their classmates, Evelyn, tried to join in their game. John pushed Sammy and Evelyn kicked him.

2. Mary and Julie were about to go into town when they realized the car had a puncture. Their next door neighbour, Peter, was working in the garden. Mary helped Julie change the wheel and Peter talked to her.

4.1.2 Crawley's Unambiguous Items with Subject Antecedent

The ambiguous items have unambiguous versions: there is only one possible antecedent that matches in gender. All 40 ambiguous items (see section 4.1.1) have an unambiguous version in which the antecedent of the pronoun is the subject of the sentence in which the pronoun appears. Note that the pronoun is still always in the direct or indirect object position.

1. John and Mary were playing in the garden. One of their classmates, Evelyn, tried to join in their game. John pushed Mary and Evelyn kicked him.

2. Mary and Tim were about to go into town when they realised the car had a puncture. Their next door neighbour, Peter, was working in the garden. Mary helped Tim change the wheel and Peter talked to her.

4.1.3 Crawley's Unambiguous Items with Object Antecedent

All 40 ambiguous items from section 4.1.1 also have an ambiguous version in which the pronoun's antecedent appears at the direct or indirect object position.

1. Mary and John were playing in the garden. One of their classmates, Evelyn, tried to join in their game. Mary pushed John and Evelyn kicked him.

2. Tim and Mary were about to go into town when they realised the car had a puncture. Their next door neighbour, Peter, was working in the garden. Tim helped Mary change the wheel and Peter talked to her.

4.1.4 Smyth's Ambiguous Pronouns in Subject Position

In Smyth (1994)'s second experiment, there are ten ambiguous items with a pronoun in the subject position. A full parallelism can be found between the subject of the item and the pronoun.

1. Mary helped Julie change the tire and then she helped Peter change the oil.

2. Shirley wrote to Carol about a meeting and then she wrote to Martin about a party.

4.1.5 Smyth's Ambiguous Pronouns in Object Position

Smyth (1994) also presents ten items with a pronoun in the direct or indirect object position. For all ten items a full parallelism can be found between the pronoun and a character in the direct or indirect object position.

1. John pushed Sammy and then Evelyn kicked him.

2. Sarah visited Cathy at home and then Charles phoned her at work.

4.2 Results

We can see in Table 3 that the model fits human preferences quite accurately. With the ambiguous items from Crawley et al. (1990) we observed that the Subject Assignment Strategy applies as a default strategy when the Parallel Function Strategy is not available. For the unambiguous items, Crawley et al. (1990) did not report human assignment. The model's assignment for these items was a 100% correct when the antecedent was a subject, but when it was an object or indirect object in

Experiment	Human		Model	
	% Sub.	% Obj.	% Sub.	% Obj.
Crawley, ambiguous items, pronoun in the object position (4.1.1)	60%	40%	72.5%	27.5%
Crawley, unambiguous items, antecedent in the subject position (4.1.2)	n.a.	n.a.	100%	0%
Crawley, unambiguous items, antecedent in the object position (4.1.3)	n.a.	n.a.	0%	85%
Smyth exp. 2, ambiguous items, pronoun in the subject position (4.1.4)	100%	0%	100%	0%
Smyth exp. 2, ambiguous items, pronoun in the object position (4.1.5)	12%	88%	30%	70%

Table 3: Human pronoun assignment versus the model's predictions on Crawley et al. (1990)'s items and Smyth (1994)'s items from experiment 2. For each item set examples can be found in section 4.1. For Crawley et al. (1990)'s unambiguous items, no human results were reported. Note that for the unambiguous items with pronouns in the object position, the model sometimes did not assign any antecedent to the pronoun.

15% of the cases the model could not attribute a score high enough to choose it as the antecedent and responded *None*[7]. For the items of Smyth (1994)'s experiment, we observed — just like him — that the Parallel Function Strategy is the preferred strategy.

4.3 Discussion

We have shown that our model is able to mirror quite accurately pronoun resolution preferences. As our model is trained on real corpus data, this means that such preferences are somehow statistically presented in the language. Our model is in line with the claim that the Parallel Function Strategy and the Subject Assignment Strategy exist alongside each other and that the former can overrule the latter. Our model embodies the idea Smyth (1994) has about pronoun resolution:

> "Pronoun resolution is a feature-match process whereby the best antecedent is that which shares the most features with the pronoun."

It also captures Smyth (1994)'s idea that not every feature has the same impact and that for example *gender match* is more important than parallel roles. Based on the results our model obtains on the experimental items, we conclude that the weights it learned from corpus are cognitively plausible.

5 Simulation of Reading Times

We use our model to simulate reading times recorded in pronoun resolution experiments. An important question is: how can our model account for those reading times? It is commonly assumed that reading time is determined by the difficulty of language processing: more difficulty will result in a longer reading time. Therefore, we need a measure of 'difficulty' from our model to simulate it. We call this measure a cost metric. In the following subsection we explain how our model can output a cost metric for pronoun resolution. We then compare our metric to reading times recorded in Crawley et al. (1990)'s experiment. [8]

5.1 Cost Metric for Pronoun Resolution

To formulate a cost metric, we have to determine first what would cause cost in pronoun resolution. We hypothesize that the difficulty of finding the antecedent is determined by the number of compatible candidates and their degree of compatibility. A higher number of compatible candidates and a higher degree of compatibility will create more competition and therefore more processing cost.

Our model is able to measure compatibility of antecedents by giving a probability score to the antecedent candidates. Nevertheless, these scores do not reflect directly the competition amongst the candidates, because the resolver makes no statements about the relation between the different scores. Therefore, to measure competition, we use the notion of entropy from Information Theory (Shannon and Weaver, 1949). Entropy is a property of a random variable and captures how much uncertainty plays a role in it. The formula of entropy — in which X is a random variable that can take the values of i — is:

$$H(X) = -\sum_{i \in X} p(X = i) \cdot \log_2(p(X = i)) \qquad (1)$$

[7] Among all antecedent candidates the correct antecedent got still the highest score, but it was lower than 50%, so the resolver responded that it did not find the antecedent. This behavior of the system can been seen as the result of training it on the Ontonotes corpus, where the bias towards classifying negative must be high, to prevent it from linking pronouns to wrong antecedents.

[8] Unfortunately, in Smyth (1994)'s experiment, no measure of processing cost was taken, so we could not apply our cost metric on its experimental items.

By defining our cost metric as the entropy over the probability distribution of antecedent candidates, we can capture the idea of competition. But a problem is that a probability distribution over antecedent candidates does not follow naturally from our model. Hence, we decided to form a probability distribution from the scores we have by using techniques inspired by Luo et al. (2004), who investigated how to form a probability distribution on entities (coreference chains) by using a probabilistic mention-pair classifier, similar to our resolver. To calculate the processing cost for a pronoun, we used the following steps:

We first get from our resolver the coreference scores between every preceding mention in the text and the pronoun to be resolved. We then group the preceding mentions by their coreference chain. Because our resolution system does not build coreference chains, this information is taken from the corpus annotation.[9] As in the work of Luo et al. (2004), each chain gets the score of its highest scoring mention. Then, among the antecedent candidates, we consider all the chains that obtain a score >0.5 [10] together with an 'empty' candidate (*i.e.* the pronoun has no antecedent) in the case that the pronoun is not anaphoric, but cataphoric.[11] We also followed Luo et al. (2004) in the assignment of probability to the empty candidate: it is given a probability equal to 1 minus the score of the highest scored mention. Next, to form a probability distribution over the mentions, we used the technique described in Luo et al. (2004): a probability distribution over the chains is formed by dividing the probability for each chain by the probability mass of all the chains in the distribution. Finally, the entropy is calculated on this distribution. This procedure is illustrated in Table 4.

m_i	P(m_i)	c_i	P(c_i)	P(dist)	Entropy
box	0.95	{*box, its*}	0.95	0.56	1.15
its	0.85				
cat	0.7	{*cat, it*}	0.7	0.41	
it	0.6				
Bob	0.01	{*Bob, he*}	0.2	–	
he	0.2				
$\emptyset$	0.05	{$\emptyset$}	0.05	0.03	

Table 4: Imagine that in a text the pronoun *it* has to be resolved and that all preceding mentions in the text are reported under m_i. First *P(m_i)* is outputted by the resolver and indicates the probability that m_i is coreferent with *it*. The empty candidate gets the score of 1 minus the highest scoring mention (hereunder: 1 - 0.95 = 0.05). Second, each mention is associated to its corefference chain c_i. Each chain gets the probability of its highest scoring mention, reported under P(c_i). Third, a probability distribution is forged from all candidates having a *P(c_i)* > 0.5 and the empty candidate. This is done by dividing the scores under *P(c_i)* by the total probability mass of the maintained candidates (hereunder: 0.95 + 0.7 + 0.05). The result is a probability distribution, reported as *P(dist)*. Entropy is calculated on this distribution.

5.2 Results

Our cost metric can mirror reading times attested in the self-paced reading experiments of Crawley et al. (1990) who reported the reading time of the last sentence of the experimental items. A significant difference was reported between the ambiguous and the unambiguous condition in an overall variance analysis of the data.[12] The model also shows this difference. When we effected an analysis of variance on a by-item basis, the factor of ambiguity was highly significant (F = 299.5, df = 1, 39, $p < .001$). In Figure 2 the predictions of the model and the actual experimental reading times are plotted against each other.

Crawley et al. (1990) also compared reading times between the subject and the object assignment in the ambiguous and the unambiguous condition. They found faster reading times for subject assignment in the ambiguous condition, but this effect only showed in an analysis by participants and not by items (F_1 = 8.52, df = 1,47, $p > 0.1$; $F_2 < 1$). They did not find significant effects in the unambiguous condition, nor in the analysis by participants, nor in the analysis by items (F_1 = 1.55, df = 1,47, $p > 0.5$; F_2 = 1.08, df = 1,39, $p > 0.5$). Like Crawley et al. (1990), our model also showed a significant difference between subject and object

[9] We make the strong assumption that recovering the coreference chains in the psycholinguistic items is rather easy and does not cause much processing cost.

[10] We do not consider mentions having scores < 0.5, because it would mean that mentions that are classified 'negative' (probability less than 50%) could be of much as an influence as candidates being classed positive. We consider that negatively classified mentions do not add much to the competition there is between antecedent candidates.

[11] Note that the pronoun cannot be expletive (*i.e.* non-referential), because this type of pronoun is not annotated as a mention in the corpus and thus not considered by the system.

[12] We do not report the F-statistic here, because only the statistics for a by-subject analysis were reported.

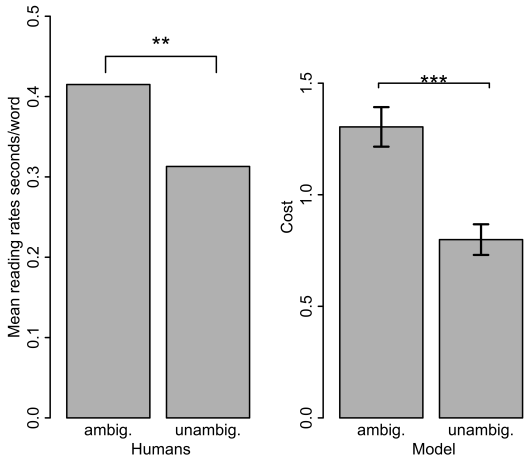


Figure 2: The model's prediction of processing cost against the reading times per word recorded by Crawley et al. (1990) for the ambiguous and the unambiguous condition of experiment 1. For the cost predicted by the model 95% confidence intervals are given. Significance codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1.

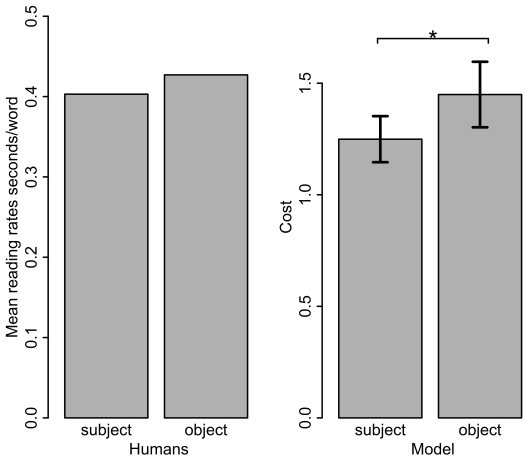
assignment in the ambiguous condition (F = 4.23, df = 1, 38, $p < .05$), but in an by-item analysis. For the unambiguous condition however, our results do not match Crawley et al. (1990)'s: we found a highly significant effect for the by-item analysis[13] (F = 24.43, df = 1, 33, $p < .001$). In Figure 3 the results for the subject and object assignment are plotted.

5.3 Discussion

Our cost metric is capable of mirroring the reading times of ambiguous versus unambiguous items and the reading times of items with subject and object antecedents in the ambiguous condition. However, in the unambiguous condition we found an effect that was not observed in the human data. We think that this can be explained by the strength of the gender and number features in our system. As the automatic gender and number feature assignment introduced some noise in our data, we think our model estimated these features lower than they should be, preventing them from erasing the influence of the Parallel Function Strategy and the Subject Assignment Strategy.

[13] In this analysis, items for which the resolver responded *None* were treated as missing values.

(a) Ambiguous Condition



(b) Unambiguous Condition

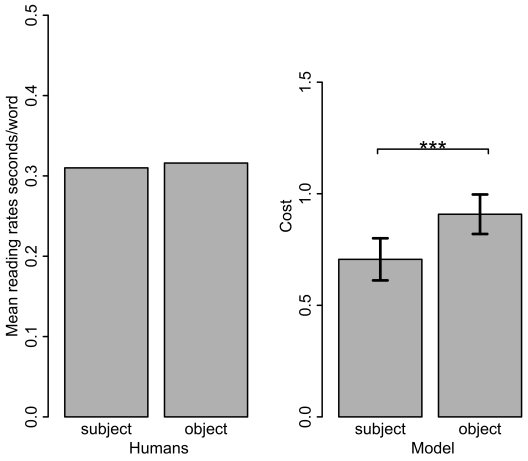


Figure 3: The model's prediction of processing cost against the reading times per word recorded by Crawley et al. (1990) for subject and object assignment in the ambiguous and the unambiguous condition of experiment 1. For the cost predicted by the model 95% confidence intervals are given.

6 General Discussion

The contribution of our model is its ability to quantify the strength of the factors of influence and its simple architecture that allows to incorporate easily new factors. The model also has the potential to explain human processing cost, because we were able to formulate a metric based on it that mirrored reading times recorded in the experiments of Crawley et al. (1990). Our results confirmed our idea that the competition between antecedent candidates can cause processing cost.

Our model can help in the psycholinguistic community to clarify statements about the exact nature of the involved factors. Indeed, when doing

the implementation of the model, many questions about how the features should be implemented arose. For example, implementing the parallel function turned out to be less straightforward than initially expected. We had to choose if we implemented it as a binary feature (the parallel function can only operate if the syntactic paths of both mentions are exactly the same), or a continuous feature (the similarity between the syntactic functions of the two mentions is what is relevant). Choices of this kind are very important when the modeling is done and inevitable. Of course, they are also relevant at the time of the design of the experimental items, but they can be overlooked more easily. The model also points out that in spite of the efforts of the experimenters to keep the items in one condition as similar as possible, many factors not included in the experimental design can still have an influence on the computational model and likely on the human participants as well. Let's take for example the items of Crawley et al. (1990): some items used proper nouns for the characters, whereas others contained only definite descriptions. This is likely to have an influence on the experienced difficulty, as suggested by the weights in Table 1, but also by theories such as the Accessibility Theory (Ariel, 1991) that states that different kinds of referential expressions are more or less accessible in memory for pronoun resolution. By detecting such things, we show that computational models can be a complementary means for psycholinguistic research.

As a future direction for our work, we plan to enhance our model, so that it would give a probability distribution over antecedent candidates in a more direct way. For the moment, as explained in section 5.1, we have to forge scores outputted by the resolver into a probability distribution, but it would be more elegant if this distribution came directly from the resolver.

We also plan to investigate further the way we define the cost metric. The idea to use entropy as a measure of uncertainty, or competition, is inspired by cost metrics for syntactic structure based on probability distribution, such as surprisal theory (Hale, 2001; Levy, 2008), predicting higher cost for unexpected syntactic structures, or the entropy reduction hypothesis (Hale, 2003; Hale, 2006), giving high cost at points where a lot of disambiguation is done. For the moment we only applied the notion of entropy, but we want to investigate if a notion of surprisal is applicable as well.

Finally, we plan to extend our model to other types of preferences. We would like for example to integrate discourse relations — that have been shown to have a great influence on pronoun resolution (Kehler and Rohde, 2013) — into our model. An even bigger challenge is to also integrate semantic information into the model. Another type of extension of our model is to get out of the experimental items and test our model on corpus data. We plan for example to test if our model can contribute to explain word by word reading times recorded on corpus — such as the Dundee eye-tracking corpus (Kennedy et al., 2003) — by adding it as a factor to a model including other factors explaining reading time, such as surprisal and word length.

7 Conclusion

In this paper we showed how a computational model can mirror human preferences in pronoun resolution and reading times with a cost metric based on the notion of entropy. We can see that the weights of the features learned on corpus correspond quite accurately to the influence of preferences in human pronoun resolution. We argue that our model will also be able to mirror other human preferences, provided we can learn the adequate features on corpus. A direction of future work is to enhance our multifactor model by more of these kinds of preferences, so that it will account for more and more preferences in pronoun resolution. We plan to ultimately test this model on reading times recorded on corpus.

Acknowledgments

We thank the three anonymous reviewers of the EACL Student Research Workshop, as well as or colleagues Tal Linzen, Maximin Coavoux and Sacha Beniamine for their comments, questions and suggestions on the paper. We also thank Adeline Nazarenko, our thesis co-director, her lab — the *Laboratoire d'Informatique de Paris Nord* — and the members of our thesis advisory committee — Saveria Colonna and Isabelle Tellier — for their support on this work. Finally, we thank our lab engineer Doriane Gras, for her help with the statistics. This work was supported by the *Labex Emperical Foundations of Linguistics* (ANR-10-LABX-0083) and the doctoral school *Frontières du Vivant*.

References

Mira Ariel. 1991. The function of accessibility in a theory of grammar. *Journal of Pragmatics*, 16(5):443–463.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 33–40. Association for Computational Linguistics.

Rosalind A. Crawley, Rosemary J. Stevenson, and David Kleinman. 1990. The use of heuristic strategies in the interpretation of pronouns. *Journal of Psycholinguistic Research*, 19(4):245–264.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Pascal Denis and Jason Baldridge. 2007. A ranking approach to pronoun resolution. In *IJCAI*, pages 1588–1593.

Amit Dubey, Frank Keller, and Patrick Sturt. 2013. Probabilistic modeling of discourse-aware sentence processing. *Topics in cognitive science*, 5(3):425–451.

Stefan L. Frank, Mathieu Koppen, Leo G. M. Noordman, and Wietske Vonk. 2007. Coherence-driven resolution of referential ambiguity: A computational model. *Memory & cognition*, 35(6):1307–1322.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

John Hale. 2003. The information conveyed by words in sentences. *Journal of Psycholinguistic Research*, 32(2):101–123.

John Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science*, 30(4):643–672.

Jerry R. Hobbs. 1976. Pronoun resolution. research report76-1. new york: Department of computer science. *City University of New York*.

Andrew Kehler and Hannah Rohde. 2013. A probabilistic reconciliation of coherence-driven and centering-driven theories of pronoun interpretation. *Theoretical Linguistics*, 39(1-2):1–37.

Frank Keller. 2010. Cognitively plausible models of human language processing. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 60–67. Association for Computational Linguistics.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The dundee corpus. In *Proceedings of the 12th European conference on eye movement*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 135–143. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Association for Computational Linguistics.

Marta Recasens and Eduard Hovy. 2009. A deeper look into features for coreference resolution. In *Discourse Anaphora and Anaphor Resolution Colloquium*, pages 29–42. Springer.

Claude E. Shannon and Warren Weaver. 1949. *The mathematical theory of communication*. University of Illinois Press.

Amy Sheldon. 1974. The role of parallel function in the acquisition of relative clauses in english. *Journal of verbal learning and verbal behavior*, 13(3):272–281.

Ron Smyth. 1994. Grammatical determinants of ambiguous pronoun resolution. *Journal of Psycholinguistic Research*, 23(3):197–229.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Rosemary J. Stevenson, Alexander WR Nelson, and Keith Stenning. 1995. The role of parallelism in strategies of pronoun comprehension. *Language and Speech*, 38(4):393–418.

Xiaofeng Yang, Jian Su, Guodong Zhou, and Chew Lim Tan. 2004. An np-cluster based approach to coreference resolution. In *Proceedings of the 20th international conference on Computational Linguistics*, page 226. Association for Computational Linguistics.

A All features that were considered before model selection

Feature	Decision
match in gender	keep
match in number	keep
m_1 is a subject	keep
match in syntactic path	keep
m_1 is a common noun	keep
m_1 is a proper name	keep
m_1 is a possessive pronoun	keep
m_1 is a personal pronoun	keep
mentions between m_1 and m_2	not significant
words between m_1 and m_2	keep
m_1 & m_2 in the same sentence	keep
length of syntactic path m_1	keep
m_1 is determined	keep
m_1 is undetermined	keep
m_1 has a demonstrative determiner	keep
m_1 spans m_2	keep
length of words of m_1	keep
number of occurrences of m_1 in the text	not significant
m_1 is a location	not significant
m_1 is a work of art	not enough data
m_1 is a geopolitical entity	keep
m_1 is an organization	not enough data
m_1 is a date	keep
m_1 is a product	not enough data
m_1 is a NORP[14]	not enough data
m_1 is a language	not enough data
m_1 is money	not enough data
m_1 is a person	not significant
m_1 is a law	not enough data
m_1 is an event	not enough data
m_1 is a quantity	not enough data

B Gender and Number Annotation

We used the Bergsma and Lin (2006) gender information, that provides counts of word forms occurring as respectively male, female and neuter gender on the web, to annotate the mentions in our corpus. More precisely, we took the three lists of unigrams (one for each gender) from the Stanford Core NLP Toolkit (Manning et al., 2014) that was compiled from the Bergsma and Lin (2006) gender information to annotate each token of a mention in our corpus with gender if it occurred in one of the lists. Then we propagated the gender of the head token to the entire mention. Finding the head of a mention was done using a heuristic: the head is the last word of the mention, except if there is a prepositional phrase inside the mention, in the latter case the head of the mention is the word before any prepositional phrase.

The number annotation was only done for tokens that were common nouns and proper names. In the tag set of the corpus singular common nouns are tagged as *NN*, singular proper names as *NNP*, plural common nouns as *NNS* and plural proper names as *NNPS*. We used these tags to assign number to tokens. Then, we proceeded with the same *head heuristic* as for the gender feature to assign number to the entire mention.

[14] nationalities, organizations, religions, and political parties

Automatic Extraction of News Values from Headline Text

Alicja Piotrkowicz
School of Computing
University of Leeds
scap@leeds.ac.uk

Vania Dimitrova
School of Computing
University of Leeds
V.G.Dimitrova@leeds.ac.uk

Katja Markert
Institut für Computerlinguistik
Universität Heidelberg
markert@cl.uni-heidelberg.de

Abstract

Headlines play a crucial role in attracting audiences' attention to online artefacts (e.g. news articles, videos, blogs). The ability to carry out an automatic, large-scale analysis of headlines is critical to facilitate the selection and prioritisation of a large volume of digital content. In journalism studies news content has been extensively studied using manually annotated news values – factors used implicitly and explicitly when making decisions on the selection and prioritisation of news items. This paper presents the first attempt at a fully automatic extraction of news values from headline text. The news values extraction methods are applied on a large headlines corpus collected from *The Guardian*, and evaluated by comparing it with a manually annotated gold standard. A crowdsourcing survey indicates that news values affect people's decisions to click on a headline, supporting the need for an automatic news values detection.

1 Introduction

In this digital age, where "the widening gap between limitless media and limited attention makes it a challenge for anything to attract an audience" (Webster, 2014), headlines play a special role. Their main function is to draw attention and act as the visual entry point to online digital content (Leckner, 2012). This is intensified on social media, where in cases of indirect engagement (e.g. with retweeted news articles) headlines are often the only visible part of the main content. Liu (2005) found that compared to print media, digital readers spend more time browsing, scanning, and keyword spotting. Various studies conducted by Chartbeat found that 38% of users leave a website immediately after accessing it[1], and that an average reader will spend only 15 seconds on a website[2]. An American Press Institute study found that roughly six in ten people acknowledge that they are "headline-gazers" checking only the headline and not reading the full article [3].

Therefore, automatic processing of headlines is needed to facilitate the selection and prioritisation of large volumes of digital content. This has been studied in the journalism field by considering news values. These are aspects of an event determining whether and to what extent it is reported, therefore guiding editorial selection. Recent journalism research (O'Neill and Harcup, 2009, p.171) suggests that news values can also be applied to the audience reception perspective, thus helping to analyse what attracts audiences to certain headlines.

The automatic extraction of news values from headlines can be a central tool for a range of applications. Automatically extracted news values scores can be correlated with online attention metrics, such as pageviews, to investigate which headline aspects influence online popularity. They can play a key role in content-based recommender systems, especially when a user model is not available (the so-called 'cold start' problem). Headline newsworthiness insights can be incorporated into online content publishing, e.g. YouTube[4] to guide authors on how to compose the headline text to attract audiences' attention. Furthermore, digital humanities researchers can conduct large-scale comparisons of news values across digital outlet types, genres, demographics, etc.

Despite the importance of headline news values, there are no automatic computational means to extract them from headline text. This requires advanced text processing to compute appropriate

[1] http://slate.me/1cJ7b5C
[2] http://yhoo.it/2cEQMVC
[3] http://bit.ly/21LwfS5
[4] https://www.youtube.com/

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 64–74, Valencia, Spain, April 3-7 2017.

features that can be related to news values. It makes for a challenging problem, because news values often involve tacit knowledge. There are no precise definitions of news values which can be used for automatic text processing, which is further aggravated by the nature of headline text. Critically, there are no studies to inform how to associate news values with various features that can be automatically extracted from headline text.

To address these challenges we utilise state-of-the-art techniques to develop a method for automatic extraction of news values from headline text. Our solution includes several NLP methods, such as wikification, sentiment analysis, and language modeling. We further combine them with other AI methods, including a burst detection algorithm to propose new techniques for estimating entities' prominence. The approach is applied and evaluated on a large corpus of news headlines from a prominent news source – *The Guardian*.

Focusing on headline news values, the paper presents a new perspective on processing digital content and contributes to text analytics by: (i) providing the first computational method for a fully automatic extraction of news values from headlines which combines relevant NLP techniques; (ii) evaluating the news values feature engineering by applying the computational method to a large corpus of news headlines and comparing the automatic annotation to a gold standard developed for this task, (iii) confirming through a user crowdsourcing study that people's choices to click on news items are influenced by news values in the headlines, indicating the significance of automatic news values detection.

2 Related Work

Headlines are gaining ground in the NLP community as a text type to be studied separately. This follows research suggesting that headlines can function autonomously from the full text. According to Dor (2003) the reader receives "the best deal in reading the headline itself". Empirical studies seem to support this – Gabielkov et al. (2016) found that 59% of shared news content on Twitter is not clicked on, i.e. has not been read before being shared. This makes headlines key for sharing content on social media. In the journalism community, the importance of headlines has already been acknowledged. For example, Althaus et al. (2001) looked at substitutes for full article text including headlines and their impact on content analysis. Tenenboim and Cohen (2013) conducted a study on the effect of headline content on clicking and commenting. However, these efforts included a manual annotation, which limited their scope. More recently, NLP researchers also focused on headlines, including headline generation (Gatti et al., 2016) and keyword selection for popularising content (Szymanski et al., 2016). We add to this ongoing NLP research by proposing news values to analyse headlines.

News values originated in the journalism studies field with the work by Galtung and Ruge (1965). Since then a variety of taxonomies of news values have been proposed: Bell (1991), Harcup and O'Neill (2001), Johnson-Cartee (2005) and Bednarek and Caple (2012). Regardless of differences in granularity and definitions, there is a considerable overlap between all these taxonomies. This allows us to select the news values which are most frequently mentioned and most relevant to headline text. These include: prominence, sentiment, superlativeness, proximity, surprise, and uniqueness. We offer a systematic and fully replicable method of an automatic extraction of these news values from headlines. Furthermore, we show that these news values influence people's decisions to click on a headline.

News values have been widely used in journalism studies, however researchers still mainly rely on manual annotation. For example, news values were used by Bednarek and Caple (2014) to analyse news discourse, while Kepplinger and Ehmig (2006) used them to predict the newsworthiness of news articles. Since news values need to be annotated manually, large-scale analyses of news articles in journalism studies have focused on aspects that are readily available through article metadata (e.g. topics in Bastos (2014)). There have been some limited attempts at using computational methods to enable large-scale annotation of news values from text, however these can be described at most as semi-automatic. For example, Potts et al. (2015) manually choose news values indicators from a preprocessed corpus; moreover, the approach relies on keywords and is topic-dependent. This paper presents the first attempt at a fully automatic and topic-independent extraction of news values which is applied and validated on headlines from a 'broadsheet' news source. Our news values detection is largely not news-specific

and can be extended to titles in other genres.

From an NLP perspective headlines pose an engineering challenge. This includes linguistic aspects like unusual use of tenses (Chovanec, 2014) and deliberate ambiguity (Brône and Coulson, 2010). There are also some domain-specific phenomena like click-baiting (Blom and Hansen, 2015). Headlines are typically short, which limits the amount of context that many NLP tools rely on. While feature engineering from headlines is less studied, there are research efforts that specifically address short texts. Tweets have attracted considerable attention, leading to the development of some Twitter-specific tools (e.g. TweetNLP[5]). Tan et al. (2014) is an example of feature engineering from tweets that looks specifically at wording and its effect on popularity. Another example of a text closely related to headlines are online content titles, e.g. image titles on Reddit (Lakkaraju et al., 2013). Many approaches include features like ratios for various parts of speech, sentiment, and similarity to a language model. However, they need to be adjusted to work with headlines. For example, since headlines offer limited context, sentiment analysis carried out on word-level is more appropriate (cf. Tan et al. (2014), Gatti et al. (2016), Szymanski et al. (2016)). For each news value we either re-implement the most appropriate state-of-the-art methods, or implement new techniques that work well with headlines.

3 Extraction of News Values

We present feature engineering methods for six news values. These six were selected, because they occur frequently in news values taxonomies (cf. Section 2). The feature computation methods are summarised in Table 2. Although our goal is a generic framework, we are inspired by research in the news domain. Consequently, the features are informed by news values related to news content.

Preprocessing. All headlines are part-of-speech tagged (Stanford POS Tagger (Toutanova et al., 2003)) and parsed (Stanford Parser (Klein and Manning, 2003)). Wikification (a method of linking keywords in text to relevant Wikipedia pages; e.g. Mihalcea and Csomai (2007)) is used to identify entities in the text. Headlines are wikified using the TagMe API[6], a tool meant for short texts, making it suitable for headlines.

[5]http://www.cs.cmu.edu/ ark/TweetNLP/

[6]http://tagme.di.unipi.it/

Notation. We see the headline H as a set of tokens obtained from the POS tagger. We denote the set of content words in H as C and the set of entities in H as E (cf. Table 1).

Table 1: Preprocessing: H (set of tokens), C (set of content words), E (set of wikified entities)

"Emma Watson's makeup tweets highlight the commodification of beauty"
H ={ *Emma*, *Watson*, *'s*, *makeup*, *tweets*, *highlight*, *the*, *commodification*, *of*, *beauty*}
C ={*makeup*, *tweets*, *highlight*, *commodification*, *beauty* }
E ={ EMMA WATSON, COMMODIFICATION }

NV1: Prominence. Reference to prominent entities (elite nations and people (Galtung and Ruge, 1965), and more recently celebrities (Harcup and O'Neill, 2001)) is one of the key news values.

We approximate prominence as the amount of online attention an entity gets. As online prominence varies with time we consider long-term vs. recent prominence and burstiness. We extend previous work by using wikification for obtaining entities and considering their burstiness.

For an entity e, we denote as $pageviews_{e,d_{-m},d_{-n}}$ the median number of Wikipedia daily page views[7] for that entity between days d_{-m} and d_{-n}. Day numbering is determined in reference to the article publication day d. Wikipedia long-term prominence is calculated over one year ($pageviews_{e,d_{-365},d_{-1}}$), and Wikipedia recent prominence on the day before publication ($pageviews_{e,d_{-1},d_{-1}}$).[8] For a news-centric perspective of prominence, we also calculate the sum of e's mentions in the news source headlines in the week before publication day, denoted as $newsmentions_{e,d_{-7},d_{-1}}$.

As entities exhibit different temporal patterns of prominence, we differentiate between entities which have a *steady* prominence (e.g. SILICONE) and entities which become *bursty*, i.e. suddenly prominent for a short period of time (e.g. EBOLA VIRUS). To identify bursty entities, we implement the burst detection algorithm by Vlachos et al. (2004) (cf. Algorithm 1). An entity is defined as *being in a burst* if its moving average in a given time frame is above the cut-off point (cf. Figure 1). We use entity bursts in two ways. Firstly, burstiness indicates the number of days that e was in a burst over a year ($daysburst_{e,d-365,d_{-1}}$). Sec-

[7]http://dumps.wikimedia.org/other/pagecounts-ez/

[8]We found the previous day's prominence to be closest to the actual on-the-day prominence.

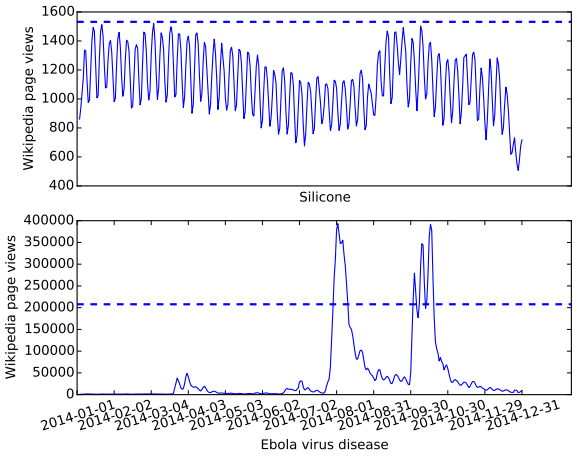


Figure 1: Time series plots of Wikipedia page views moving averages (MA) for two entities: non-bursty SILICONE (top) and bursty EBOLA VIRUS DISEASE (bottom). The dashed line shows the burst cut-off line.

ondly, current burst size indicates how many standard deviations above MA_e is any e which is in a burst day before publication ($daysburst_{e,d-1,d-1}$ returns 1 if e is in a burst, 0 if not). We are the first to consider burstiness for popularity prediction.

Algorithm 1 Burst detection algorithm adapted from Vlachos et al. (2004). Following experimentation, moving average was set to three days and the cut-off point to two times standard deviation.

1: Calculate moving average of length 3 for entity e (MA_e) for sequence $d_{-365}, ...d_{-1}$.
2: Set cutoff = $mean(MA_e) + 2 \times SD(MA_e)$
3: Bursts = $d_i | MA_e(i) > cutoff$

As a headline can have multiple entities, all prominence measures are aggregated via summation over all entities in H (see Table 2).

NV2: Sentiment. This refers to sentiment-charged events (Johnson-Cartee, 2005) and using sentiment-charged language (Bednarek and Caple, 2012). Features relating to sentiment and emotionality have been shown to influence a news article's virality (Berger and Milkman, 2012). However, this effect has not been studied for headlines.

As direct measures of sentiment, we combine SentiWordNet (Baccianella et al., 2010) positivity and negativity scores of content words, and calculate sentiment and polarity scores following Kucuktunc et al. (2012). Sentiment can also be indirect. Firstly, a word may be in itself objective, but carry a negative connotation (e.g. *scream*). We therefore measure the percentage of content words in a headline with a positive or negative connotation (using a connotations lexicon (Feng et al., 2013)). Secondly, we measure the percentage of biased content words (using a bias lexicon (Recasens et al., 2013)). For example, the same political organisation can be described as *far-right*, *nationalist*, or *fascist*, each of these words indicating a bias towards a certain reading.

NV3: Superlativeness. The size (Johnson-Cartee, 2005, p.128), or magnitude (Harcup and O'Neill, 2001) of an event is considered to influence news selection.

We focus on explicit linguistic indicators of event size: comparatives and superlatives (indicated by part-of-speech tags), and amplifiers (indicated with intensifiers and downtoners). For the latter, we combine the lists in Quirk et al. (1985) and Biber (1991), obtaining wordlists of 248 intensifiers and 39 downtoners.

NV4: Proximity. This news value has been interpreted as both geographical (Johnson-Cartee, 2005, p.128) and cultural proximity (Galtung and Ruge, 1965) of the event to the news source or the reader (Caple and Bednarek, 2013).

Following an assumption that readers from the country of a news outlet constitute the main part of its readership, we focus on geographic proximity to the news source. We use a binary feature that indicates whether a headline refers to an entity that is geographically close to the news source, and manually create a wordlist including names for the country, regions, capital city (17 UK-related terms in total). We then look for matches in the headline text ("*London* smog warning as Saharan sand sweeps southern *England*") or the Wikipedia categories of each entity supplied in the TagMe output (category POSTAL SYSTEM OF THE UNITED KINGDOM for headline "Undervaluing *Royal Mail* shares cost taxpayers £750m in one day").

NV5: Surprise. Events which involve "surprise and/or contrast" (Harcup and O'Neill, 2001) make news. Surprise in headlines can be implicit ("Denver Post hires Whoopi Goldberg to write for marijuana blog"), which requires world knowledge to identify it, or explicit ("Beekeeper creates *coat of living bees*"), where it arises from unusual word combinations.

We target explicit surprise by calculating the commonness of phrases in headlines with reference to a large corpus. We first extract phrases of following types: SUBJ-V, V-OBJ, ADV-V, ADJ-

Table 2: Feature implementations and statistics on *The Guardian*. Notation is in Table 1. Measures: median and maximum values, prevalence (proportion of non-zero scores), and the Kruskall-Wallis test comparing the manual gold standard to automatic extraction (* $p<0.05$, ** $p<0.01$, *** $p<0.001$).

	Feature name	Implementation	Median	Max	Prevalence	KW
NV1	number of entities	$\|E\|$	1	8	79%	***
	Wikipedia current burst size	$\sum_{e \in E} daysburst_{e,d_{-1},d_{-1}} \times \frac{pageviews(e,d_{-1},d_{-1}) - \text{mean}(MA_e)}{\text{SD}(MA_e)}$	0	57.16	12%	0.2
	Wikipedia burstiness	$\sum_{e \in E} daysburst_{e,d_{-365},d_{-1}}$	21	156	78%	***
	Wikipedia long-term prominence	$\sum_{e \in E} pageviews_{e,d_{-365},d_{-1}}$	1,342	125,757	79%	***
	Wikipedia day-before prominence	$\sum_{e \in E} pageviews_{e,d_{-1},d_{-1}}$	1,642	1,031,722	78%	***
	News source recent prominence	$\sum_{e \in E} newsmentions_{e,d_{-7},d_{-1}}$	0	122	50%	**
NV2	sentiment	$max\_positivity - max\_negativity - 2$	-2	-1	100%	0.1
	polarity	$max\_positivity + max\_negativity$	0.5	1.88	79%	**
	connotations	$\frac{\text{\# content words with positive or negative connotations}}{\|C\|}$	0.34	1	92%	0.2
	bias	$\frac{\text{\# biased content words}}{\|C\|}$	0.13	1	61%	*
NV3	comparative/superlative	$\frac{\text{\# words with JJR\|JJS\|RBR\|RBS POS tag}}{\|C\|}$	0	1	7%	***
	intensifiers	$\frac{\text{\# intensifiers}}{\|H\|}$	0	0.34	10%	***
	downtoners	$\frac{\text{\# downtoners}}{\|H\|}$	0	0.29	4%	0.2
NV4	proximity	1 if explicit reference to UK in H or in Wikipedia category tags, else 0	0	1	35%	***
NV5	surprise	$min\text{LL}_p$ where LL_p is the log-likelihood for a phrase in H	4.15	2,726,186	100%	*
NV6	uniqueness	$max_{t \in d-72hr}$ cosine similarity$(H, pastH_t)$	0	0.83	13%	*

N, N-N; and generate a regular expression with their inflected forms (e.g. *man drinks* → *man drinks|drank|drinking*). For each regexp we obtain a count from a Wikipedia corpus[9] and sum the counts for each phrase and calculate its log-likelihood (LL). The feature value is given the lowest LL in the headline (as we are looking for the most surprising phrase)[10].

NV6: Uniqueness. News has to be new – "any new comment or circumstance [...] adds to the debate" (Conley and Lamble, 2006). An analysis of several storylines in the headlines corpus showed that of two very similar headlines, the latter tends to be less popular ("Ferry disaster: South Korean prime minister resigns" was more popular than the later "South Korean prime minister resigns over ferry sinking").

For a headline H we select past headlines from 72 hours before H's publication and which have at least one TagMe entity overlapping or neither has any entities[11]. For a pair of H and $pastH$ vectors (created using a *tf-idf* weighted Gigaword corpus) we calculate their cosine similarity. The highest cosine similarity is assigned as the feature value.

[9] http://www.nlp.cs.nyu.edu/wikipedia-data

[10] We experimented with other corpora and metrics and found Wikipedia and log-likelihood to give best results.

[11] Entity overlap helps with ensuring that the headlines are part of the same storyline; including headlines with no entities ensures more coverage. Collecting headlines from previous 72 hours works better than other cutoff points.

4 Application and Evaluation

We applied the feature extraction methods on a corpus of headlines from *The Guardian*, a major British newspaper. This provides a wide coverage of various topics and genres, allowing a good exploration of news values. The automatic extraction of news values was compared to a manually annotated gold standard.

Headline corpus. The headlines corpus was built using the Guardian Content API[12]. We downloaded all headlines published during April 2014, yielding a corpus of 11,980 headlines.

Automatic annotation. Feature values were calculated for each headline. Statistics for the extracted features in *The Guardian* corpus are reported in Table 2 (Median, Max, Prevalence).

Manually annotated gold standard. For each news value we selected 20 headlines from the headlines corpus. In order to use the clearest examples for a more accurate annotation, we randomly selected 10 headlines from the top quartile values and 10 from the bottom quartile. For news values that are split into multiple features (NV1:Prominence, NV2:Sentiment, NV3:Superlativeness), the feature group vectors were ordered to obtain quartiles. Overall, a total of 120 headlines were selected for manual annotation. Three expert annotators, PhD students in linguistics, annotated each headline as positive or

[12] http://www.theguardian.com/open-platform

negative (Y/N) for the first five news values (cf. Table 3). For NV6:Uniqueness, annotators were presented with 20 headlines from the corpus and further 20 past headlines with highest and lowest headline uniqueness scores (which were randomly sampled). The annotators indicated whether any of the past headlines were very similar (i.e. highly related) to a given headline.

Inter-annotator agreement. The inter-annotator agreement was calculated using Fleiss's Kappa. It ranges from substantial for NV1:Prominence (.76) and NV6:Uniqueness (.73), through moderate for NV3:Superlativeness (.43), NV5:Surprise (.48), and NV4:Proximity (.55), to fair for NV2:Sentiment (.22). The annotators remarked that sometimes they chose 'on instinct' and their responses might vary from day to day. This highlights the challenge of an automatic detection of news values, as news values are somehow tacitly understood. The annotators' judgments were aggregated using a majority vote, creating the gold standard.

Comparison with gold standard. We calculated pairwise comparisons between each feature and the relevant manual label (e.g. number of entities and Prominence, bias and Sentiment). The Kruskal-Wallis test was used to determine whether the differences in feature values for the two manual annotation labels (Y/N) were significant (cf. column KW in Table 2). These results indicate whether the value calculated for a given feature correctly reflects the presence of a news value in the gold standard produced by the human experts. The findings of the evaluation are discussed below.

5 Discussion of Feature Extraction

We use a news corpus that is representative of a wide range of news publications under the umbrella of 'broadsheet' (as opposed to tabloid newspapers which differ in style and tone). *The Guardian* corpus is a freely available resource, allowing replication of methods and study findings. While the evaluation of feature extraction is conducted over one corpus, we also applied this approach to another publicly available 'broadsheet' corpus – *New York Times* (cf. Appendix A). We will discuss below the findings from *The Guardian* evaluation study, and will refer to feature extraction outputs from *New York Times* to illustrate feature behaviour on two corpora.

NV1: Prominence is one of the most prevalent news values and our approach using wikification proves very reliable. It occurs quite frequently – most headlines in *The Guardian* corpus have at least one entity (median number of entities = 1), which attracts a fair amount of online attention (median Wikipedia long-term prominence = 1,342 pageviews). Some headlines include very prominent entities (max. Wikipedia day-before prominence = 1,031,722). The outputs from *New York Times* are similar – every headline is associated with at least one Wikipedia entity (100% prevalence for number of entities); and Wikipedia burstiness, long-term, and day-before prominence have non-zero scores in 66% of headlines. This shows that Wikipedia provides a wide coverage for the computation of prominence. Wikipedia current burst size is a rare feature (12% in *The Guardian* and 10% in NYT), because capturing an entity in a burst is uncommon, since bursts do not apply to all entities and do not happen frequently.

The IAA for Prominence is the highest (κ=.76) and nearly all features reach $p<0.001$ when compared to the manual annotations. This strongly supports our implementation of Prominence, in particular the use of wikification and Wikipedia as a prominence source. Burstiness presents a new way of looking at Prominence. While burstiness (i.e. how many times in a year an entity had pageviews significantly higher than its average) is a reliable feature, current burst size (i.e. size of the burst on the day before article publication) is not significantly correlated with the gold standard.

NV2: Sentiment is among the most challenging news values to implement, since it is not typical for broadsheets and sentiment-charged language in headlines does not always accurately reflect the true sentiment or emotion. Headlines in broadsheet newspapers tend to be quite neutral (median sentiment = -2; median polarity = 0.5). This is also the case for the *New York Times* (sentiment = -2; polarity = 0). However, most headlines contain at least one connotated or biased word (connotations prevalence = 92%, bias prevalence = 61%; slightly lower in NYT: 78% and 51%).

The IAA was fair, at κ=.22. The fact that many headlines are neutral can explain the low agreement, since the neutral cases are where experts are more likely to disagree. Furthermore, while manual annotation for one aspect of Sentiment like positivity/negativity can achieve substantial agreement (.76 agreement between experts in Snow et

Table 3: Examples of annotated headlines. Y/N: majority vote manual annotation. Below: automatically extracted values aggregated via summation by feature group (cf. Table 2 for feature value ranges).

#	Headline	Prominence	Sentiment	Superlativeness	Proximity	Surprise
E1	"Getting really hung up on EE/Orange customer service"	Y 0	Y 3	Y 0.125	Y 0	Y 3.23
E2	"Mount Everest avalanche leaves at least 12 Nepalese climbers dead"	Y 13272	Y 4.25	Y 0.17	N 0	N 4.15
E3	"Huzzah for foreign experts. After all, they're better than our own"	N 672	Y 2.75	Y 0.2	N 0	Y 398
E4	"Rev; Martin Amis's England; and A Very British Renaissance: TV review – video"'	Y 36236	N 2.45	N 0.08	Y 1	N 4.15
E5	"This week's new live comedy'	N 0	N 3.25	N 0	N 0	N 102

al. (2008)), our definition of Sentiment is broader. The annotators pointed out an interesting characteristic of expressing Sentiment. On one hand, there were highly evocative headlines that describe some tragic news events (+sentiment, +emotion). On the other hand, some headlines use sentiment-charged language, but were not evocative to the same extent (+sentiment, -emotion). For example, *comedy* (E5 in Table 3) has positive sentiment, but does not evoke positive emotion. When compared to the manual annotations, two out of four Sentiment features reach significance levels, so our implementation does capture some aspects of Sentiment. Extracting Sentiment from headlines proves a challenge, since they are short texts with limited context and often the sentiment is implied or requires world knowledge to identify (e.g. "Guinea's Ebola outbreak: what is the virus and what's being done?"). Disentangling sentiment and emotion might paint a clearer picture.

NV3: Superlativeness is rare, but reliably extracted. It is the least prevalent news value (between 4-10%; between 3-6% in NYT). The median values are also all zero. Our narrower definition of could be the reason, however we decided to focus on explicit linguistic indicators of event size (e.g. *very*, *hardly*) to keep the implementation topic-independent and more easily generalisable.

The IAA was moderate (κ=.43). Two out of three features were significant at $p<0.001$. This confirms that our approach that relies on POS tags and wordlists does capture this news value. The only feature not to reach a significance level was downtoners. Downtoners are a class of words which aim to diminish the word they describe (e.g. *nearly*, *barely*, *just*). They are not only rare (prevalence is 4%), but also require specific knowledge to identify them (we identified 39 downtoners, compared to 248 intensifiers). Bearing in mind that downtoners might have more impact if their coverage increases with a more comprehensive wordlist, the other Superlativeness features (comparative/superlative and intensifiers) can be reliably used for headlines.

NV4: Proximity is not frequent, but our approach using a wordlist and Wikipedia categories proves very reliable. This news values occurs in 35% of headlines. This is not surprising, considering that *The Guardian* has a global audience, so the majority of news is not UK-specific (prevalence in NYT is similar at 32%).

The IAA is moderate (κ=.55). The feature reaches significance at $p<0.001$, so our method of capturing Proximity is well-supported. Using entity categories ensures wider coverage and less manual effort than just using a wordlist. This is turn depends on the reliability of the NER/wikification tools. In some cases an entity might be missed (cf. E1 in Table 3, where *EE/Orange* was missed and consequently both Prominence and Proximity scores are zero). It is important to note that Proximity covers both geographic and cultural proximity. Our annotators were UK residents, familiar with *The Guardian*, but demographics of the reader will probably influence their familiarity with some entities. In our future work we will include some demographics data to deepen the implementation for Proximity.

NV5: Surprise is difficult to implement due to peculiarities of headline text, but our approach which targets surprising phrasing using a Wikipedia-based language model does capture it. The median log-likelihood for this features is relatively low (4.15; 4.04 for NYT), which means that most headlines have fairly surprising phrasing. This might be because headlines do not tend to strictly follow the conventions of everyday language (e.g. frequent use of untensed verbs and

noun clusters). When using a corpus which is not specifically for headlines (we used Wikipedia), the log-likelihood will tend to be lower.

The IAA was moderate at κ=.48 and the feature is significant ($p<0.05$). This shows that using a count-based method captures this news value. In other genres where surprise might play a bigger role, this method can be extended by using a headline-specific corpus or building language model that takes into account syntactic structure.

NV6: Uniqueness, or rather a lack of it, is fairly rare, but our implementation reliably identities such instances. The prevalence is quite low (15%; but slightly higher at 34% in *New York Times*), which follows the basic journalistic principle that news have to be novel.

IAA was substantial with κ=.73 and the feature was significant ($p<0.05$), so we can be sure that any similar headlines are identified. An analysis of headlines with non-zero Uniqueness values reveals that most of them are either part of a regular feature (e.g. "Reviews roundup"), or part of continuing storylines about the same event (often featuring some media like video).

Overall, the results of the evaluation are encouraging: for every news value the majority of features significantly differentiates between the manual annotation labels. This means that our approach successfully identified and quantified at least some aspects of every news value.

The study also indicated open issues requiring further investigation. Firstly, the findings highlight the importance of world knowledge when analysing headlines. For example, for the well-established NLP topic like sentiment analysis, we find that although purely linguistic methods can capture most phenomena in headlines, they fall short to recognise sentiment within entities (e.g. Ebola). Similarly, a more generic approach for Proximity would require world knowledge to detect that an entity is related to the reader's location. We are addressing this in our future work. Secondly, it will be interesting to explore how the proposed methods can be applied to other types of news sources (e.g. tabloids) and to genres other than news. With the exception of news source prominence and uniqueness, our features are not news-specific. Titles for other types of digital content (blogs, videos) also include prominent entities, sentiment or intensifiers. News values detection offers a new perspective for their analysis. Thirdly, our methods can be adapted to other languages, provided that certain NLP resources exist (POS tagger, NER, sentiment lexicon). This would enable large-scale analyses of headlines along multiple axes, like language and genre.

6 Do News Values Influence People's Choice of Headlines?

To show the importance of the automatic news value extraction for a range of applications (cf. Section 1), we examined whether news values matter for general audiences. This was explored with a crowdsourcing study.

Survey content. The survey consisted of five short sections for news values NV1 to NV5 (NV6:Uniqueness was not included, because we decided to focus on news values which are expressed within a single headline, whereas the Uniqueness feature requires comparing headlines). In each section participants were presented with a short definition and several examples. Then they were asked the following: *"I personally consider this news value when clicking on headlines"* and given five Likert scale responses (cf. Figure 2). Standard demographics information (age, gender, country of residence, native language, news reading habits) was collected.

Participants. The crowdsourcing platform CrowdFlower was used to recruit participants for the survey, allowing us to collect responses globally, thus reflecting the global nature of audiences of online news outlets. The survey took approximately 10 minutes to complete and participants were paid $2 for taking part. Out of 100 collected responses, 96 were recorded as complete. While quality of responses was generally quite high, we carried out some quality control. We removed any responses where more than 75% of answers were neutral, as well as responses where time to complete was in the bottom quartile (to ensure that participants had taken time to understand the concepts). After the quality control measures, 71 responses were selected: 48 participants were 34 or younger and 23 were 35 or older; 17 were female, 54 were male; 30 were native English speakers and 41 were non-native English speakers; 44 participants read news daily, 27 weekly.

Results and discussion. Results are presented in Fig. 2. The overall impact that news values have on survey participants has been indicated as very positive. NV1:Prominence, NV4:Proximity,

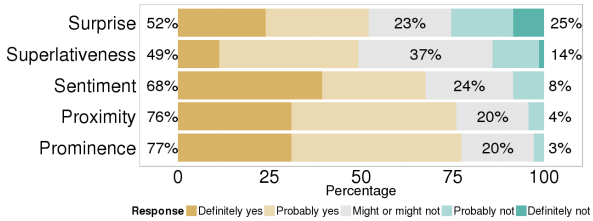


Figure 2: Survey results to the question *"I personally consider this news value when clicking on headlines"* (N=71). Percentages show aggregated positive, neutral, and negative responses.

and NV2:Sentiment had the highest proportions of positive answers (77%, 76%, and 68%, respectively). This follows the journalism studies literature, where these three news values attract perhaps the most focus. Comparison with the gold standard confirmed that our implementation for NV1:Prominence and NV4:Proximity reflects the experts' judgments. Since this survey highlighted the role of Sentiment, we are motivated to develop it further to capture its full extent. NV3:Superlativeness had the most neutral responses (37%). On one hand, this could be because this news value is slightly more difficult to understand[13]. On the other hand, Superlativeness might have been deemed to play a lesser role, since its main function is more supportive (to embellish or diminish content). Finally, NV5:Surprise had the most negative responses (25%). This might be because surprising headlines could be perceived as less informative, or more ambiguous. As people often read only headlines to get their news (Gabielkov et al., 2016), surprise would not support the headlines' function as summaries.

Overall, results of this survey highlight the importance of news values in headlines. We also found that news values play a role for both native and non-native speakers of English (our sample has roughly equal numbers of both). This is important, since most major news outlets nowadays have a more global reach.

7 Conclusions and Future Work

The work presented here is the first step in a larger project to predict the popularity of news articles using headlines. Our focus on headlines is motivated by their role in the everyday online experience, characterised by limited audience attention and the frequent use of social media websites.

We proposed an automatic extraction method for *news values*, which have been posited in journalism studies and offer a new perspective on characterising digital content. We broke novel ground by developing fully automatic and topic-independent methods for identifying news values in headlines. An evaluation using manual annotations shows that for all news values the output of the automatic extraction corresponds to the gold standard. The results from a crowdsourced survey indicated that news values influence people's decisions to click on a headline. This supports the wider adoption of the automatic method of analysing headlines in a range of applications concerning human choices (e.g. prediction models, recommender systems, intelligent assistants).

Our current and future work includes several stages. Firstly, we have collected a second corpus (*New York Times*) to apply our news values extraction methods. Secondly, the extracted news values scores are being correlated with popularity of headlines on social media and applied in a popularity prediction model using machine learning methods. The results from the manual annotations and the crowdsourced survey will also be used to inform the weights of features in the prediction model. Furthermore, another survey will target the direct engagement with headlines (i.e. whether a reader would click the headline) and compare it to the social media popularity metrics we have already collected. Finally, using both data from the crowdsourced surveys and publicly available Twitter data we will look at whether demographics, in particular the country of residence, have impact on the news values of Prominence and Proximity. We will use the data on the entities we identified from knowledge bases like *Wikidata* and *BabelNet* to enrich the implementations of these news values.

Acknowledgments

This work was supported by a Doctoral Training Grant from the Engineering and Physical Sciences Research Council. Data collection and storage comply with EPSRC data management policies. The dataset is available at `http://doi.org/10.5518/147`.

We would also like to thank our expert annotators for their work and feedback.

[13] 57% of native English speakers judged Superlativeness positively compared to 44% of non-native speakers.

References

Scott L. Althaus, Jill A. Edy, and Patricia F. Phalen. 2001. Using substitutes for full-text news stories in content analysis: Which text is best? *American Journal of Political Science*, 45(3):pp. 707–723.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Marco Toledo Bastos. 2014. Shares, pins, and tweets: News readership from daily papers to social media. *Journalism Studies*, pages 1–21.

Monika Bednarek and Helen Caple. 2012. *News Discourse*. Continuum.

Monika Bednarek and Helen Caple. 2014. Why do news values matter? Towards a new methodological framework for analysing news discourse in Critical Discourse Analysis and beyond. *Discourse & Society*, 25(2):135–158.

Allan Bell. 1991. *The language of news media*. Blackwell Oxford.

Jonah Berger and Katherine L. Milkman. 2012. What makes online content viral? *Journal of Marketing Research*, 49(2):192–205.

Douglas Biber. 1991. *Variation across speech and writing*. Cambridge University Press.

Jonas Nygaard Blom and Kenneth Reinecke Hansen. 2015. Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics*, 76:87–100.

Geert Brône and Seana Coulson. 2010. Processing deliberate ambiguity in newspaper headlines: Double grounding. *Discourse Processes*, 47(3):212–236.

Helen Caple and Monika Bednarek. 2013. Delving into the discourse: Approaches to news values in journalism studies and beyond. *Reuters Institute for the Study of Journalism*.

Jan Chovanec. 2014. *Pragmatics of Tense and Time in News: From canonical headlines to online news texts*. Pragmatics & Beyond New Series. John Benjamins Publishing Company.

David Conley and Stephen Lamble. 2006. *The Daily Miracle: An Introduction to Journalism*. Oxford University Press.

Daniel Dor. 2003. On newspaper headlines as relevance optimizers. *Journal of Pragmatics*, 35(5):695–721.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1774–1784, Association for Computational Linguistics.

Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. 2016. Social Clicks: What and Who Gets Read on Twitter? In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pages 179–192, ACM.

Johan Galtung and Mari Holmboe Ruge. 1965. The structure of foreign news the presentation of the Congo, Cuba and Cyprus Crises in four Norwegian newspapers. *Journal of Peace Research*, 2(1):64–90.

Lorenzo Gatti, Gözde Özdal, Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2016. Automatic creation of flexible catchy headlines. In *Natural Language Processing meets Journalism Workshop*. IJCAI.

Tony Harcup and Deirdre O'Neill. 2001. What is news? Galtung and Ruge revisited. *Journalism Studies*, 2(2):261–280.

Karen S. Johnson-Cartee. 2005. *News narratives and news framing: Constructing political reality*. Rowman & Littlefield Publishers.

Hans Mathias Kepplinger and Simone Christine Ehmig. 2006. Predicting news decisions. An empirical test of the two-component theory of news selection. *Communications*, 31(1):25–43.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Association for Computational Linguistics.

Onur Kucuktunc, Berkant Barla Cambazoglu, Ingmar Weber, and Hakan Ferhatosmanoglu. 2012. A large-scale sentiment analysis for Yahoo! answers. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 633–642, ACM.

Himabindu Lakkaraju, Julian J. McAuley, and Jure Leskovec. 2013. What's in a Name? Understanding the Interplay between Titles, Content, and Communities in Social Media. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, pages 311–320

Sara Leckner. 2012. Presentation factors affecting reading behaviour in readers of newspaper media: an eye-tracking perspective. *Visual Communication*, 11(2):163–184.

Ziming Liu. 2005. Reading behavior in the digital environment: Changes in reading behavior over the past ten years. *Journal of documentation*, 61(6):700–712.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, ACM.

Deirdre O'Neill and Tony Harcup. 2009. News values and selectivity. *The Handbook of Journalism Studies*, pages 161–174.

Amanda Potts, Monika Bednarek, and Helen Caple. 2015. How can computer-based methods help researchers to investigate news values in large datasets? *Discourse & Communication*, 9(2):149–172.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A comprehensive grammar of the English language*. Longman.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659, Sofia, Bulgaria, Association for Computational Linguistics.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Association for Computational Linguistics.

Terrence Szymanski, Claudia Orellana-Rodriguez, and Mark T. Keane. 2016. Helping news editors write better headlines: A recommender to improve the keyword contents and shareability of news headlines. In *Natural Language Processing meets Journalism Workshop*. IJCAI.

Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 175–185, Baltimore, Maryland, June. Association for Computational Linguistics.

Ori Tenenboim and Akiba A. Cohen. 2013. What prompts users to click and comment: A longitudinal study of online news. *Journalism*, 16(2):198–217.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180.

Michail Vlachos, Christopher Meek, Zografoula Vagena, and Dimitrios Gunopulos. 2004. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 2942.

James G. Webster. 2014. *The marketplace of attention: How audiences take shape in a digital age*. MIT Press.

A Supplementary Material: Feature Extraction on *New York Times*

Table 4: Feature extraction statistics on *New York Times* corpus. Notation is explained in Table 1. Reported measures: median and maximum values, prevalence (proportion of non-zero scores). WP=Wikipedia.

Feature name	Median	Max	Prevalence
NV1: Prominence			
Number of entities	1	4	100%
WP current burst size	0	57.18	10%
WP burstiness	15	166	66%
WP long-term prominence	626	65,327	66%
WP day-before prominence	773	467,458	66%
News source recent prominence	0	70	32%
NV2: Sentiment			
Sentiment	-2	-1	100%
Polarity	0	1.88	43%
Connotations	0.25	1	78%
Bias	0.11	1	51%
NV3: Superlativeness			
Comparative/superlative	0	1	3%
Intensifiers	0	0.33	6%
Downtoners	0	0.33	3%
NV4: Proximity			
Proximity	0	1	32%
NV5: Surprise			
Surprise	4.04	2,724,886	100%
NV6: Uniqueness			
Uniqueness	0	1	34%

Assessing Convincingness of Arguments in Online Debates with Limited Number of Features

Lisa Andreevna Chalaguine
Department of Computer Science
University College London
United Kingdom
ucablc3@ucl.ac.uk

Claudia Schulz
Department of Computing
Imperial College London
United Kingdom
claudia.schulz@imperial.ac.uk

Abstract

We propose a new method in the field of argument analysis in social media to determining convincingness of arguments in online debates, following previous research by Habernal and Gurevych (2016). Rather than using argument specific feature values, we measure feature values relative to the average value in the debate, allowing us to determine argument convincingness with fewer features (between 5 and 35) than normally used for natural language processing tasks. We use a simple forward-feeding neural network for this task and achieve an accuracy of 0.77 which is comparable to the accuracy obtained using 64k features and a support vector machine by Habernal and Gurevych.

1 Introduction

Argumentation is the foundation of reasoning, no matter in what discipline: if someone wants to publish scientific discovery, evidence to support the discovery is required; reasoning in law uses argumentation to solve legal disputes, and political debaters adopt informal logics and argumentation to achieve approval with the voting population (Boltuzic, 2013). Being an important element of human communication and being frequently used in texts, argumentation has attracted significant research focus from many disciplines, ranging from philosophy to artificial intelligence (Goudas et al., 2014).

Initially, argument mining focused on specific domains such as legal texts (Palau and Moens, 2009) and scientific publications, social media being a much less explored domain. However, argument mining and analysis in online content has gained significant interest in the last couple of years. Since an increasing portion of information and opinion exchange occurs in online interactions on social media, it is a valuable domain for gaining understanding of the reasons underpinning users' opinions (Snajder and Boltuzic, 2014). Suitably mined and analysed, it could provide a lot of insight into the beliefs and reasoning of people about problems that are affecting our society (Wells, 2014) such as public opinion on political decisions, cultural issues and historical events.

The aim of argument mining is to extract arguments and their relations from text to then use argumentation frameworks to evaluate which arguments "win" a debate (Cerutti et al., 2014; Cerutti et al., 2016; Abdallah et al., 2010). However, especially in online interactions on social media, some arguments are better than others, so different arguments should have different intrinsic strengths, and there are indeed various argumentation frameworks which assume that arguments have intrinsic strengths (Leite and Martins, 2011; Rago et al., 2016). Thus, it is important to evaluate the strength of arguments, which we do in terms of convincingness, following the work of Habernal and Gurevych (2016).

The problem of argument analysis in informal domains such as social media, however, is the vagueness, implicity and wordiness (taking more words than necessary to make your point) of the users' arguments and the characteristics of natural dialogue in general as opposed to formalised debates and structured documents (Concannon et al., 2015). Apart from being a highly subjective task by itself, discrepancies of quality amongst platforms and even individual discussions are significant: one argument that is *convincing* in one debate is not necessarily convincing when placed in another, even if it is on the same topic. Therefore, it is important to take the overall quality of the

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 75–83, Valencia, Spain, April 3-7 2017.

given debate into account, when judging whether a specific argument is convincing or not.

Habernal and Gurevych (2016) cast the problem as relation classification, where a pair of arguments having the same stance to the same subject are compared and labeled by human annotators as either the first argument being *more* convincing, or the first argument being *less* convincing than the second. They use two machine learning methods for predicting the relation of an argument pair: a feature-rich support vector machine (SVM) and a bidirectional long-short-term memory neural network (BLSTM) with pre-trained word vectors. They achieve an accuracy of 0.78 and 0.76, respectively.

Our study focuses on the same task, however, since the argument pairs are created per debate, we believed that feature values used to determine the convincingness should be relative to that whole debate. Therefore, instead of extracting a large amount of features for each argument independently, we calculate an argument's features with relation to the *average argument* of the debate, thus taking into account that convincingness is relative to the debate, rather than absolute. This allows us to consider a much smaller amount of features than normally used for natural language processing (NLP) tasks.

The paper is structured as follows: Section 2 describes the data set that was used for the experiments and evaluation. Section 3 introduces the algorithm used to calculate the feature vectors of arguments and the experimental setup. Section 4 describes and analyses the results and compares our approach to Habernal's and Gurevych's (2016). Section 5 points out some of the limitations of our approach and finally, Section 6 presents our conclusions and outlines future research.

2 Data Set

Since the objective of our work is the same as Habernal's und Gurevych's (2016), we used their newly created corpus of annotated argument pairs, measuring convincingness. It is constructed from 32 debates about 16 topics taken from *createdebate.com* and *procon.org* and contains 16k argument pairs. An argument is a single comment posted by a user (and will be used in this context throughout the rest of this paper). An argument pair is a set of two arguments belonging to the same debate. From each topic 25-35 random arguments were sampled and *(n * (n – 1)/2)* argument pairs created by combining all selected arguments. Those argument pairs are labeled as to which one is more convincing[1] and each of the annotated argument pairs comes with five textual reasons that explain the annotator's decision since assessing convincingness of a single argument directly is a highly subjective task with high risk of introducing bias due to personal beliefs, preferences and background (Habernal and Gurevych, 2016).

3 Methodology

3.1 Feature Selection

Early implementations of NLP tasks usually involved the hand-coding of large sets of rules. Modern NLP algorithms are largely based on statistical machine learning. The machine-learning paradigm instead uses learning algorithms like statistical inference in order to automatically learn such rules through the analysis of large corpora (Chopra et al., 2013). These algorithms take as input a set of *features* that are extracted from the given input data.

There are many state of the art features that are very popular and often used for argument mining and other NLP tasks. Those include word mean length, discourse marker count, named entities (NE), part-of-speech (POS) tags, readability measurements and punctuation. Apart from those we also used surface features like number of sentences, number of words and average number of words per sentence. We also counted the most common unigrams (words), bigrams and trigrams, long words and the average frequency distribution of the words in an argument, spelling mistakes, hyperlinks and rude words. Regarding punctuation and digits we counted the number of question marks, exclamation marks, full stops, percentage signs and numbers. We selected the features according to what we believed could contribute to the convincingness of an argument. For example we chose *number of hyperlinks*, because some users back up their arguments with references to websites. Lists of common words were created because we assumed that the most common words of a debate would give a good indication of what the debate was about. Therefore, an argument that included some of those words, has a high chance of

[1] neither we, nor Habernal and Gurevych considered arguments which were labeled as *equally convincing*

Algorithm 1 Debate Feature Extraction

```
1:  procedure DEBATEFE(wholeDebate)
2:      arg_counter = 0
3:      debate = []
4:      for i do in range (1, wholeDebate.end)          ▷ iterate through whole debate
5:          argument = argument.i
6:          debate = debate + argument
7:          arg_counter += 1
8:      debate_length = length(tokenise(debate))        ▷ number of words
9:      debate_nrSent = length(sent_tokenise(debate))   ▷ number of sentences
10:     average_length = debate_length / arg_counter
11:     average_nrSent = debate_nrSent / arg_counter
12:     [...]                                           ▷ more feature extraction (e.g. avg sentence length etc.)
13:     preprocess(debate)                              ▷ deleting stop words, POS-tagging, stemming
14:     most_common_stems = extract_mc_stems(debate)
15:     most_common_nouns = extract_mc_lemmas(debate)
16:     [...]                                           ▷ more NLP feature extraction (e.g. most common bigrams, trigrams etc)
```

being relevant for this particular debate. In total we analysed 35 features[2].

3.1.1 Examples

3.2 Calculation of Vector Values

Since we wanted to put the features of the individual arguments in relation with each other, we needed to obtain values to compare those features against. This was done by extracting features from the whole debate first, and then comparing the features from the individual arguments against those. We therefore used a simple but effective method as shown in Algorithm 1. We concatenated all arguments of a debate into one *single text* and extracted from it the features mentioned above: calculating the average of the feature for the debate (e.g. average number of words per argument, average number of sentences etc.) and creating lists of *most common (MC) words*. Then we extracted the same features from the individual arguments and calculated the ratio of the individual metrics to the previously calculated average, making the individual feature value relative to the average debate value. For example, the length feature would be calculated like this:

$$\textbf{length ratio} = \frac{\textbf{length of ind. argument}}{\textbf{length of avg. argument}}$$

And the feature *intersection (IS) of the most common (MC) words ratio* would be calculated as follows:

$$\textbf{IS MC words ratio} = \frac{|\textbf{MC words debate} \cap \textbf{MC words argument}|}{|\textbf{MC words debate}|}$$

Thus, if the average argument length in a debate was 5 sentences and the individual argument was 4 sentences, the arguments length (in sentences) feature would be 0.8. If the MC-word list contained 10 words and the individual argument mentioned 4 of them, the arguments MC-word feature would be 0.4. This method was used for lemmas, stems, nouns, bigrams and trigrams.

We also extracted certain independent features where the values were not compared against the debate average (e.g. number of insulting words or number of exclamation/question marks), because we wanted to see whether such "unique" features had an impact on the convincingness of the argument.

All those values were then used to create the feature vector of the argument. This makes our approach domain independent, however it puts the arguments within a debate into relationship with the other arguments and treats them in context rather than evaluating them independently and out of context.

3.3 Analysis via Forward-Feeding Neural Network

After creating the individual feature vectors, the vectors of both arguments were concatenated in order to represent an argument pair - a total of 70 features. The first 35 being the features of the first argument and the next 35 being the features of the second argument. These vectors were fed into a

[2] for a detailed description of the 35 features and their calculation see http://www.homepages.ucl.ac.uk˜ucablc3/img/report.pdf

Group	Features	Accuracy
Group I	length ratio (words)	75%
Group II	length ratio (sentences); IS MC lemmas and stems ratios	65-70%
Group III	percentage of long words; IS MC nouns ratio	60-65%
Group IV	percentage of misspelled words; percentage of long rare words	55-60%
Group V	avg. no. of words per sentence; avg. sentence length ratio; percentage of discourse markers; no. of rude words; capscount; digits; percent signs; NE ratio; percentage of MC nouns, lemmas, stems and bigrams; IS MC bigrams ratio; percentage of unusual words	50-55%
Group VI	avg. length of word; readability; no. of hyperlinks; percentage of adjectives and adverbs; avg. rarity of words	50%
Group VII	punctuation count; percentage of nouns and pronouns; percentage of MC trigrams; IS MC trigrams ratio	<50%

Table 1: Feature Groups and the averaged accuracy of the individual features in that group. If the word *ratio* is used, it means it is calculated against the debate's average, if the word *percentage* is used, the value was calculated against the individual argument only

simple feed-forward neural network (FFNN) with one hidden layer. The number of nodes in the input layer is *features * 2*, the hidden layer has two nodes, as has the output layer. We trained the FFNN with the ADAM optimiser (Kingma and Ba, 2014) as Habernal and Gurevych did for their BLSTM, however instead of binary cross entropy we used a logistic regression cross entropy loss function which is commonly used when using a softmax layer as the final layer. The reason for choosing a softmax output layer (instead of a sigmoid layer like Habernal and Gurevych did) was that the outputs sum up to 1 and therefore represent probabilities for the convincingness of each argument. We round the predictions to get the outcome *1,0* if the first argument is more convincing than the second and *0,1* if the second argument is more convincing than the first, hence the two output neurons. We use sigmoid as an activation function (Habernal and Gurevych do not mention what they used as an activation function).

3.4 Individual feature testing

Since we were interested in what features would give the best results in order to identify the most relevant for predicting which argument was more convincing, we first trained the FFNN with each feature individually to see its impact on the accuracy of prediction. We divided the data into two sets of equal size and trained the neural network on each set[3], using the other set as the test data and averaged the results. The worst prediction was as low as 48% for the percentage[4] of *nouns* in an argument (*number of nouns/number of words*) and the highest one was 75% for the length ratio (in words).

The features were then divided into 7 groups as shown in Table 1, grouping the ones with similar accuracy together within 5% ranges, starting at 45%. In the three highest groups, ranging from 60% to 75%, were six features, namely: the length ratios (in words and in sentences), the IS MC lemmas, stems and nouns ratios, and percentage of long words (minimum 10 characters). Only the last feature is independent and counts the number of long words in each argument without considering the whole debate (*number of long words/number of words*).

3.5 Combination of feature groups

We expected that by combining different features with each other we could obtain an even higher accuracy across all the debates. Therefore, we combined the features in one group as well as different feature groups with each other to see how the results change. For this setup (and all following experiments) we used the same approach as Habernal and Gurevych, namely *cross validation* and tested on each individual debate, using all debates but one as training data and the particular debate as testing data. This setup made it possible to establish which features were more relevant for which debate and how they influenced each other, as well as speculating the underlying reasons of the results obtained. The average accuracy for each feature group combination as well as the average of the individual features included in those groups can be seen in Table 2. For the accuracy of features combined all features were used during testing. The average of the individual features was calculated by averaging the accuracies of each individually tested feature in that particular group. The higher accuracies for the combined features shows that using features of similar individual ac-

[3] we did not use cross validation due to the time constraints of the project

[4] percentage is used for independent features when considering individual argument only

Combination	Accuracy of Features Combined	Avg of Ind. Features
Group I	75.87%	**75.87%**
Group II	71.50%	66%
Group III	64.96%	62.75%
Group IV	60.53%	57.25%
Group V	60.50%	51.88%
Group VI	59.18%	50.00%
Group VII	50.89%	49.25%
Groups I,II	75.84%	68.38%
Groups I,III	76.42%	66.83%
Groups II,III	76.48%	64.80%
Group I,II,III	**76.57%**	66.50%
Group I,II,III,IV	76.38%	64.19%
Group IV,V,VI	67.34%	51.91%
Group I,II,III,IV,V,VI	76.24%	55.08%
Group I,II,III,IV,V,VI,VII	75.42%	54.20%

Table 2: Combinations of Feature Groups that were tested and their accuracies used combined as a group as well as the average of the individually used features

curacies together, gives more accurate results.

4 Results

4.1 Evaluation

Table 2 shows that combining features that independently have a similar accuracy, can achieve an up to 9% higher accuracy when used together. Using as many features as possible may therefore seem like an effective strategy. However, when combining the different feature groups together, we can observe that after a certain point, adding more features that resulted in a lower accuracy, has a negative impact on the overall accuracy. Using all features gives the worst result and we can conclude that even though *highly relevant* features are included, the *less relevant* features influence the result in a negative way.

The most successful feature is the length ratio (in words), as already observed during the individual feature testing. Combining it with Group II, which represents the IS MC lemmas and stems ratios and the length ratio in sentences, results in almost the same, however slightly lower accuracy. From this follows that it is not necessary to consider the most common words in an argument that is longer than the average and/or longer than the one compared against (the other length ratio likely does not make a difference because of the previously measured one). An intuitive explanation for these results is that the length of an argument might be an indicator that it is better explained and/or more informative. The presence of common words in the debate, on the other hand, might not be an indicator of convincingness, especially if the argument is introducing new ideas (therefore probably new words). For example, in the debate for banning plastic bottles, the three most common words[5] are *water, plastic* and *bottle*. Since the debate is about *plastic bottles* it is not surprising that those words are mentioned in an argument. All annotators agreed that in the following two arguments, the first one is more convincing because it is more informative, although both of them use two of the three most common (lemmatised) words, namely *water* and *bottle*.

(1) In New York City alone, the transportation of bottled water from western Europe released an estimated 3,800 tons of global warming pollution into the atmosphere. In California, 18 million gallons of bottled water were shipped in from Fiji in 2006, producing about 2,500 tons of global warming pollution.

(2) Bottled water is not strictly regulated while tap water is, so you have no idea what you are drinking when you drink bottled water.

The length ratio feature (in words) combined with the IS MC nouns ratio and percentage of long words in the argument (Group III), however, increases accuracy by almost 1 percent. We explain this as follows - although the Group III features have a lower accuracy on their own than Groups I and II, if any of the Groups I and II features are already *given* (like the length or/and a big intersection of the most common words) the presence of long words in the argument makes it qualitatively even better. Especially if it also mentions the most common nouns in the debate which ensures that it is not off-topic and certainly relevant. This is because long words have a higher information content resulting in the argument being more informative and therefore likely to be more convincing (Piantadosi et al., 2011). The first argument shown above indeed contains two long words (minimum 10 character), namely *transportation* and *atmosphere* as well as the most common noun *water*. Now, given Groups I and III, adding Group

[5] words were lemmatised in order to avoid counting the same word with different endings

Debate	Stance	FFNN	SVM	BLSTM
Ban Plastic Bottles	Yes	89%	85%	76%
	No	85%	90%	88%
Atheism vs	A	80%	81%	80%
Christianity	C	70%	68%	75%
Creation vs	C	81%	84%	88%
Evolution	E	62%	65%	77%
IE vs Firefox	IE	77%	84%	81%
	FF	83%	82%	78%
Gay Marriage	Right	76%	76%	74%
	Wrong	85%	82%	87%
Should parents use	No	80%	84%	78%
spanking?	Yes	77%	79%	68%
If spouse	No	72%	71%	64%
committed murder...	Yes	77%	79%	72%
India to lead	No	77%	82%	77%
the world	Yes	71%	69%	79%
Be fatherless or	F	77%	77%	69%
have a lousy father	LF	70%	67%	60%
Is porn wrong	No	77%	82%	79%
	Yes	81%	85%	85%
School Uniform	Bad	74%	75%	78%
	Good	83%	83%	74%
Abortion	Pro	68%	71%	68%
	Contra	78%	79%	80%
PE mandatory	No	79%	79%	80%
	Yes	77%	79%	78%
TV or Books	TV	80%	78%	73%
	Books	76%	78%	75%
Common Good vs	CC	72%	72%	78%
Personal Pursuit	PP	67%	67%	68%
Farquhar founder	No	71%	79%	63%
of Singapore	Yes	84%	85%	76%
Average		**77%**	**78%**	**76%**

Table 3: Result Comparison between our Feed-Forward Neural Network (FFNN) and Habernal and Gurevych's Support Vector Machine (SVM) and Bidirectional Long Short-Term Memory Neural Network (BLSTM)

II increases accuracy even further, seemingly because arguments that are longer, contain the most common words and nouns of the debate, as well as long words are the most convincing. The average result of the Groups I, II and III is 76.57%. As soon as we add other feature groups to this combination, accuracy decreases.

Nevertheless, it should be mentioned that including the Group IV features significantly increases the accuracy of certain debates (up to 4%). This is the case for debates where the overall quality of the discussion is lower and arguments tend to be not very long. Long words, especially if not previously mentioned in the debate and grammar errors in such debates are therefore a better indicator for judging whether an argument is considered as *convincing* or not.

The results presented lead to the conclusion that, although we normalised the length features, the unnormalised ones would have given the same results, since the longer one of two arguments is always ranked as "more convincing". Therefore, we do not have to calculate the average length of an argument in a given debate and can use the unnormalised values of the arguments length, average sentence length and percentage of long words, together with the most common stems and noun ratios.

4.2 Comparison with existing work

Habernal and Gurevych use a SVM (as a "traditional" model) which they train with different NLP features, including, uni- and bigram presence, adjective and verb endings, contextuality measures, ratio of exclamation and punctuation marks, ratio of modal verbs, POS-tags, past- and future tense verbs, many different readability measures, five sentiment scores, spell checking and surface features like sentence length, longer words etc. ending up with vectors of size 64k.

They also use a BLSTM neural network that they train with word embeddings from Global Vectors[6]. The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. For large corpora, this pass can be computationally expensive, but it is a one-time up-front cost. For training they used 840B tokens from Common Crawl[7]. As a consequence that this method is not domain independent and depends on the features of the corpora that were used for obtaining the word embeddings.

Table 3 shows our 5-feature[8] vector results using our FFNN compared to the SVM and BLSTM used by (Habernal and Gurevych, 2016). The average accuracy of our FFNN is only one percent below the SVM which was trained with vectors containing over 64k features.

Habernal and Gurevych claim that both of their tested systems outperform simple baseline lemma n-gram presence features with SVM which only performed 65%. In the individual feature testing phase, using only the *IS MC lemmas ratio* fea-

[6]http://nlp.stanford.edu/projects/glove/

[7]http://commoncrawl.org/

[8]because including stems, lemmas or both had no impact on the results we included stems only in our "top feature set" because they are less expensive to compute

ture resulted in 66% in our case. We do not know how many features Habernal and Gurevych used for their baseline.

SVM and NN often get quite similar results if the same parameters are used (Romeo and Toppo, 2006). The SVM support vectors are equivalent to the weights of the NN. It is therefore not the choice of machine learning tool that is responsible for the results but the choice of parameters and their weights/support vectors. As Habernal and Gurevych observe themselves - feature extraction for SVM requires heavy language-specific preprocessing machinery and favour BLSTM because it "only" requires pre-trained embedding vectors. However, this is slightly misleading since the vectors also need pre-training (even though it being a one-time cost) and training requires suitable corpora. Our approach, on the other hand, does not require exhaustive NLP preprocessing of the given data and the accuracy is not dependent on any pre-trained vectors where the choice of why that specific corpora was used for training might not be very transparent.

The main difference between our approach and the SVM used by Habernal and Gurevych is that they analyse each argument individually and extract general features independent of other arguments in that particular debate. We, on the other hand, based on the assumption that the convincingness of an argument is context dependent, extract general features of the whole debate first, and calculate the value of the individual features relative to the previously calculated average for that feature. Four of our five best performing features are debate-dependent and only one is an independent one. This is the reason why we need only five features to get similar results as Habernal and Gurevych got using a vector dimension of over 64k. As mentioned above, despite judging whether an argument is *convincing* is a highly subjective task, and although we have for now eliminated the problem of comparing different stances - the overall quality of the debate is still highly relevant and has to be taken into account when deciding which argument is more convincing. The percentual representation of the divergence from the debate's *average* argument is a much more representative metric when analysing qualitatively different debates than using the actual number of words, sentences, POS-tags etc. for each individual argument.

4.3 Discussion

As mentioned previously, analysing online content is a fairly new field of research, which currently makes use of methods that are mainly used for *argument extraction* from "professionally" written texts like articles and academic papers. In order to extract argumentative structures out of a structured text, a large amount of linguistic features are required. Analysing comments in an online debate, where each comment is treated as one argument, however, is a very different task that requires a different approach. One could still look for argument structures and try to extract the premise and the conclusion, however, in online debates like those represented in the corpus, it is questionable how accurate those results would be due to noise and the informality of online-language. If the whole debate is quite "primitive", extracting advanced NLP features might prove counter productive. Instead of intensive analysis, that is unlikely to lead to much better results, we therefore propose simple and light general features that can be extracted quickly and cheaply and results in accuracies up to almost 90% in the best and only as low as 65% in the worst debates.

4.4 Ranking

Currently, to the best of our knowledge, there are no implemented methods in forums or other social media that are able to identify the best or worst arguments in a debate or dialogue. Arguments (or posts) are most commonly ranked by other users depending whether they agree with the stance that the argument supports, like on *debate.org* or (usually on product reviews) whether they found the particular review *helpful* or not, a typical example being product reviews on *Amazon*. As mentioned before - no matter how low the quality of the debate is, there will still always be one argument that is the "best" in this particular debate. Using our method could help to identify the *best* ones without the user having to read through everything himself.

In order to evaluate whether the accuracy predictions of our neural network could be used to perform such a task, we created rankings for certain debates by counting how many times each argument was labeled *more convincing* by the annotators and sorted them accordingly - the argument which was voted *more convincing* most often being the first/best. We then compared this ranking

to the ranking that was obtained by the predictions of our neural network. We analysed six debates, the two with the highest prediction accuracy, the two with the lowest and two average debates. In the debate with the highest prediction accuracy, only 4 out of 24 arguments had a rank-difference of 3 to 4 places, the rest were ranked either exactly the same or with a rank difference of 1 place. In the debate with the lowest prediction accuracy only 9 out of 30 were correctly ranked. Interestingly the difference in ranking accuracy between the debate with the second lowest and the second highest prediction accuracy is not as significant as one would expect. This is because the difference in prediction accuracy might be caused by one single argument that confused the neural network and was always wrongly labeled as *more convincing*, while the rest were labeled correctly. If, for example, in a debate with 5 arguments (ranked 1, 2, 3, 4, 5), which results in 10 argument pairs and therefore 10 comparisons, argument 1 was labeled wrongly once against argument 2, the prediction accuracy would be 90% and the resulting ranking 2, 1, 3, 4, 5. If argument 1 was labeled wrongly against arguments 2, 3 and 4 the prediction accuracy would be significantly lower, namely 70% and the ranking 2, 3, 4, 1, 5. However, we would still extract the *top four* arguments in the debate.

In 5 out of the 6 analysed debates (including the worst one) our neural network correctly predicted the *top five* arguments of the debate.

5 Limitations

Despite the high prediction accuracy for certain debates, the low accuracy for other debates shows that the current approach is still far from complete (see Table 3). The reasons include:

Low predictions for certain debates:
The low accuracy is due to reasons that are not easily caught by simple features. Those include the detection of sarcasm and passive aggression and poor and unclear sentence structure. More sophisticated and costly features are needed, however, more research needs to be conducted in order to identify what sort of features and methods are suitable for this sort of domain.

Low accuracy of certain features:
For the NLP feature extraction we use off-the-shelf classifiers that are not always accurate like, NLTKs[9] POS-tagging and NE-extraction, because we did not train them for a social media domain. Training POS-taggers and NE-extractors ourselves could lead to better results and therefore increase accuracy of those features.

Results very corpus specific:
For now, our results can only be judged against Habernal's and Gurevych's who used (and created) the same corpus. Like for all supervised machine learning research, more labeled data would be required to test the generality of our approach. It would be interesting to take a debate that developed on social media or a news website and analyse results.

6 Conclusion and Future Work

We have shown that a small number of features can be enough to predict the convincingness of an argument in social media discussions compared to existing approaches that use a very large feature set or extensive machine learning training, if those features are calculated in relation to the whole debate. The corpus created by Habernal and Gurevych (2016) was used for the experiments and their results were used for comparison. We used a simple machine learning method, namely a feed-forward neural network, using a small but well picked number of features for predicting the convincingness of arguments that are analysed in pairs. We extended Habernal's and Gurevych's study (2016) with a detailed analysis of linguistic and general features and explanations of their impact on the accuracy of the prediction. We then used our observations to hand-pick the features with the highest accuracy which resulted in a total vector dimension of 10 (*2 * 5*) instead of 64k as used by them for their support vector machine and achieved almost the same results. Out of the five best performing features four follow our novel idea of feature values relative to the average argument and only two require some sort of natural language processing, namely a POS-tagger for extracting nouns and a word-stemmer. Our code is freely available on github[10].

We would like to point out that in order to make claims about the general applicability of our method for determining convincingness of argu-

[9] http://www.nltk.org/
[10] https://github.com/lisanka93/individualProject

ments, more data is required[11]. It should also be noted that the annotator's classification of certain argument pairs is debatable. This is not surprising, since even annotators disagreed on some of those and an argument was labeled as "more convincing" if three out of five annotators agreed. However, our study proves that, given the corpus of Habernal and Gurevych, only a fraction of the amount of features used by their SVM is necessary to solve the task at hand.

In the future it would be of interest to see if this approach of using feature values relative to the debate is also useful for other classification tasks in argument mining, for example classifying the relation between arguments as attacks or supports. It would also be interesting to see whether one could measure the overall *stance* or *emotion* of the debate and compare it to the individual arguments.

7 Acknowledgments

We thank our colleague Oana Cocarascu from Imperial College London who provided insight and expertise that greatly assisted the research, as well as Luka Milic for assistance with the implementation of the neural network.

References

Sherief Abdallah, Ruqiyabi Naz Awan, Jean-Franois Bonnefon, Mohammed Iqbal Madakkate, , and Iyad Rahwan. 2010. Behavioral experiments for assessing the abstract argumentation semantics of reinstatement. *Cognitive Science 34, no. 8*.

Filip Boltuzic. 2013. Computational approaches to argumentation in natural language text. *Faculty of Electrical Engineering and Computing, University of Zagreb, Ph.D. proposal*.

Federico Cerutti, Nava Tintarev, and Nir Oren. 2014. Formal arguments, preferences, and natural language interfaces to humans: An empirical evaluation. *Proceedings of the 21st European Conference on Artificial Intelligence*.

Federico Cerutti, Alexis Palmer, Ariel Rosenfeld, and Francesca Toni. 2016. A pilot study in using argumentation frameworks for online debates. *Proceedings of the First International Workshop on Systems and Algorithms for Formal Argumentation*.

Abhimanyu Chopra, Abhinav Prashar, and Chandresh Sain. 2013. Natural language processing. *International Journal of Technology Enhancements and Engineering Research, vol 1, issue 4*.

[11]To the best of our knowledge the corpus by Habernal and Gurevych is the only corpus on the convincingness of arguments

Shauna Concannon, Patrick Healey, and Matthew Purver. 2015. How natural is argument in natural dialogue? *eecs.qmul.ac.uk*.

Theodosis Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. 2014. Argument extraction from news blogs and social mediat. *Artificial Intelligence: Methods and Applications*.

Ivan Habernal and Iryna Gurevych. 2016. Which argument is more convincing? analysing and predicting convincingness of web arguments using bidirectional lstm. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations 2015*.

Joo Leite and Joo Martins. 2011. Social abstract argumentation. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argument mining: The detection, classification and structuring of arguments in text. *Twelfth International Conference on Artificial Intelligence and Law*.

Steven Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimised for efficient communication. *Proceedings of the National Academy of Sciences*.

Antonio Rago, Kristijonas Cyras, and Francesca Toni. 2016. Adapting the df-quad algorithm to bipolar argumentation. *Workshop on Systems and Algorithms for Formal Argumentation at COMMA*.

Enrique Romeo and Daniel Toppo. 2006. Comparing support vector machines and feed-forward neural networks with similar parameters. *International Conference on Intelligent Data Engineering and Automated Learning*.

Jan Snajder and Filip Boltuzic. 2014. Back up your stance: Recognising arguments in online discussions. *Proceedings of the First Workshop on Argumentation Mining*.

Simon Wells. 2014. Argument mining: Was ist das? *Proceedings of the 14th International Workshop on Computational Models of Natural Argumen*.

Zipf's and Benford's laws in Twitter hashtags

José Alberto Pérez Melián
Univ. Politècnica de València
Camí de Vera s/n, 46022
València, Spain
jopeme@inf.upv.es

J. Alberto Conejero
IUMPA-UPV
Univ. Politècnica de València
Camí de Vera s/n, 46022
València, Spain
aconejero@upv.es

Cèsar Ferri
DSIC
Univ. Politècnica de València
Camí de Vera s/n, 46022
València, Spain
cferri@dsic.upv.es

Abstract

Social networks have transformed communication dramatically in recent years through the rise of new platforms and the development of a new language of communication. This landscape requires new forms to describe and predict the behaviour of users in networks. This paper presents an analysis of the frequency distribution of hashtag popularity in Twitter conversations. Our objective is to determine if these frequency distribution follow some well-known frequency distribution that many real-life sets of numerical data satisfy. In particular, we study the similarity of frequency distribution of hashtag popularity with respect to Zipf's law, an empirical law referring to the phenomenon that many types of data in social sciences can be approximated with a Zipfian distribution. Additionally, we also analyse Benford's law, is a special case of Zipf's law, a common pattern about the frequency distribution of leading digits. In order to compute correctly the frequency distribution of hashtag popularity, we need to correct many spelling errors that Twitter's users introduce. For this purpose we introduce a new filter to correct hashtag mistake based on string distances. The experiments obtained employing datasets of Twitter streams generated under controlled conditions show that Benford's law and Zipf's law can be used to model hashtag frequency distribution.

1 Introduction

Twitter is a microblogging social network launched in 2006 with 310 million active users per month and where 340 million tweets are daily generated[1]. By sending short messages called tweets of up to 140 characters, users can insert text, pictures, videos and links to interact with other users over the network. Twitter users can interact between them by using the @ symbol followed by the username they want to mention. They can also classify tweets in more than one category or theme by using *hashtags* (alphanumeric strings preceded by #). Hashtags are created by users. Some of them propagate and thrive while others are restricted to a few mentions and die. The most popular hashtags reach out what is called the trending topic list, who shows the most popular hashtags used at the moment. Popularity is considered either at a local level or worldwide. In this sense, the authors of (Ma et al., 2012) present a method to predict hashtag success. Hashtags are extremely popular in Twitter. Some studies have analysed how to extract hashtags from a microblogging environment (Efron, 2010). Other works apply Diffusion of Innovation (DoI) to model hashtag life cycle (Chang, 2010). However, to the best of our knowledge, there are not studies about the frequency distribution of hashtag popularity in Twitter conversations. In this work, our goal is to analyse Twitter datasets in order to discover if the the frequency of hashtags popularity follow some of the distribution laws that are very common in many types of data presented in the social sciences. Specifically, we study Benford's law and Zipf's law.

Benford's law (Benford, 1938), also known as the first-digit law, characterises the distribution of digits in large datasets. This law takes into account that in many natural occurring systems the frequency of number's first digits is not evenly dis-

[1] https://about.twitter.com/company

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 84–93, Valencia, Spain, April 3-7 2017.

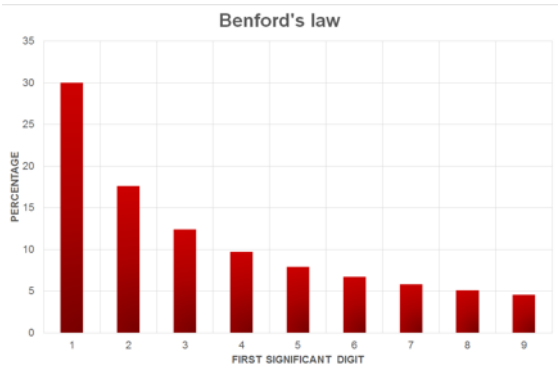


Figure 1: First Significant Digit probabilities calculated by Benford's law

tributed. Benford observed that numbers with 1 as first digit were observed far more often than those starting with 2, 3 and so on. The probability *P* of a number *d* having a particular non-zero first digit is given by formula 1.

$$P(d) = \log_{10}\left(1 + \frac{1}{d}\right) \quad (1)$$

For instance: if we have the number 81291, the First Significant Digit (FSD) is 8, the second is 1 and so on. Figure (1) shows the probabilities for the first significant digit distribution. The probability to find a 1 in the first position is about 30%, while the probability to find a 9 is around 4.6%. Some authors have applied Benford's law to forensic account (Durtschi et al., 2004), where an anomalous data distribution in the first significant digits can lead to detect fraud. It has been also applied to social networks by counting friends and followers distributions in Facebook, Twitter and many more networks (Golbeck, 2015). Other fields where Benford's law has been applied are: crime statistics (Hickman and Rice, 2010), electoral fraud (Bërdufi, 2013; Battersby, 2009), genome data (Friar et al., 2012) and macroeconomic data (Müller, 2011). For a recent account on other computer approaches for studying social networks, we refer the reader to (Kurka et al., 2016).

A related empirical law is Zipf's law. In fact, Benford can be seen as a special case of Zipf's law. Zipf confirmed that given a corpus with word frequencies of a language, the frequency of each word is inversely proportional to its position in the ranking of word's frequencies, see an updated reference in (Zipf, 1949). Both ranking and frequency distributions follow an inverse relationship who can be approximated by formula (2), where P_n represents the frequency of a word sorted in the *n*-th position with the exponent *a* very near to 1. Some applications of Zipf's law can be seen in (Powers, 1998; Popescu, 2003; Huang et al., 2008).

$$P_n \sim \frac{1}{n^a} \quad (2)$$

In this work we have considered, as corpus sets, hashtags appearing in some collection of tweets. The frequency in which they appear coincides with the number of times every tweet is mentioned. Therefore, in order to test Zipf's law on each dataset, we rank hashtags in the order from most to least relevant. For carrying out these analysis we have considered two different datasets that are described in Section 2. These datasets are processed in Section 3 in order to bring together hashtags with certain plausible typesetting mistakes or that were expected to refer to the same topic. Additionally, we also have optimised the process of joining similar hashtags in every dataset in order to drastically reduce computing times. Once the frequency of every hashtag is computed, in Section 4 we analyse the distribution of these frequencies in order to test whether Zipf's and Benford's law are satisfied. Conclusions are reported in Section 5.

2 Data Extraction

In this section, we summarise the process of collecting and extracting the datasets that is going to be employed in the experiments. Tweets of the datasets have been downloaded by means of the twitter API service [2]. This API provides programmatic access to Twitter data. Tweets are extracted in JSON format, and in every tweet we can find 26 different features [3]. In this work we only employ the field *["entities"]["hashtags"]* that contains the list of hashtags mentioned on the tweet and help us to count the total number of mentions of hashtags in a dataset.

The code for the use of Twitter API functions as well as for the data management has been developed in Python. This programming language provides a huge set of libraries for API connection and data management.

After we get the complete list of hashtags included in the dataset, we need to standardise and

[2] https://dev.twitter.com/overview/api

[3] https://dev.twitter.com/overview/api/tweets

	Users	Tweets	With #	Unique #
Argentina	650	635765	89643	44235
Chile	650	625739	63387	60262
Colombia	650	616046	144352	52248
Spain	650	623670	176167	76762
Mexico	650	624161	138631	66955
Peru	650	621325	144561	65156
Venezuela	650	610692	173906	59839
Total	4550	4357398	930647	425457

Table 1: Information about dataset *Hispatweets*. Number of users, number of tweets, number of tweets that contain hashtags and number of distinct hashtags.

normalise it in order to analyse correctly the hashtag distribution. The first step of this process consists in converting all the text in lower case characters. Given that the analysed tweets are in Spanish, we need to avoid the confusion that accents and some of the letters of the Spanish alphabet could produce[4]. Concretely, we remove accents and diaeresis from vowels, and the character *ñ* is converted into *n*.

In this work, we use two different datasets: *Hispatweets* and *Elecciones*. In the following points we summarise the information about these datasets.

2.1 Dataset *Hispatweets*

The dataset *Hispatweets* contains tweets from seven countries where different types of Spanish is spoken: Argentina, Chile, Colombia, Spain, Mexico, Peru and Venezuela. This dataset was generated in order to study the different features of the Spanish that is used in Twitter in each one of these countries. For that goal, 650 users of each country were selected and a set of tweets generated by these users were downloaded. Information about the creation of this dataset can be found in (Fabra-Boluda, 2016). The dataset is available in the following url: `https://s3.amazonaws.com/cosmos.datasets/hispatweets-populated.zip`.

In Table 1 we include some information about this dataset. In total, there are 4357398 tweets distributed almost uniformly among the seven countries. The presence of hashtags in the tweets is not uniform. Spain is the country where tweets contain more hashtags, since 21.36% of the tweets have at least one hashtag. The last column contains the number of different hashtags after the standardisation process.

[4]Some users tend to avoid the use of accents in Twitter hashtags.

Hashtag	Users
#PartidoPopular	Mariano Rajoy - @marianorajoy
	Soraya Saenz - @Sorayapp
#Ciudadanos	Albert Rivera - @Albert_Rivera
#PSOE	Pedro Sánchez - @sanchezcastejon
#Podemos	Pablo Iglesias - @Pablo_Iglesias_
#IzquierdaUnida	Alberto Garzón - @agarzon

Table 2: Hashtags and users employed in the dataset *Elecciones*.

2.2 Dataset *Elecciones*

The dataset *Elecciones* is formed by tweets collected during the 2015 Spanish General Election campaign on December 2015. Specifically, the tweets were stored during the period of the election campaign that started on 1/12/2015 and finished on 22/12/2015. For every day in this period, a Python script was executed every eight hours to download tweets referring some hashtags related to the main parties and tweets mentioning political leaders that were involved in the electoral process. Table 2 shows the exact terms that were explored for extracting the tweets. Summing up, this dataset is formed by 256293 tweets that contain 171650 hashtags (7950 distinct hashtags are distinguished).

3 Hashtag identification

After removing special characters from the hashtags, we observed that most of them had a low number of mentions, in many cases due to spelling errors on them. For instance: the hashtag *#7deldebatedecisivo* used for one of the debates for the 2015 Spanish General Election had a high number of mentions. Around them we find with hashtags like *#7ddebatedeviscisivo* or *#7deldevate* who had few mentions (both containing spelling errors).

For studying distributions of hashtags mentions in Twitter conversations, it is important if we are able to detect and correct in some way this kind of problems in hashtag identification. One possibility could be the use of automatic spell checkers in order to detect and correct spelling mistakes. Nevertheless, this solution is not feasible in this context for some reasons. Mainly because hashtags usually concatenate words, and strings without separators between the words are ambiguous and cannot be parsed correctly in many cases. This problem has been defined in NLP as compound

splitting (Srinivasan et al., ; Koehn and Knight, 2003). Additionally, in many cases hashtags contain acronyms, slang words or proper nouns, and these are not easily identified by compound splitting techniques and spell checkers.

Given these limitations, we have adopted a different approach based on the similarity of hashtags. We assume that in many cases if two hashtags are very similar (i.e, the similarity between the two terms is above a certain threshold α), they can be joined to be accounted as the same term. Therefore we need to measure similarities between terms. There is a plethora of different metrics that allow to estimate the distance between strings (Cohen et al., 2003). We have applied three string distances, *Levenshtein* distance, *Jaro* distance and *Jaro-Winkler* distance. These measures are implemented in the *python-Levenshtein*[5] library, written in Python. For a detailed description of these string distance metrics we refer to (Naumann and Herschel, 2010). A comparison between the differences in their application can be found in (Cohen et al., 2003). In this work we have used four levels for α: 0.95, 0.90, 0.85 and 0.80. Using smaller values can lead to group hashtags that are not very similar among them.

Table 3 shows an example of the measures of the string distances applied to some hashtags. Note that a measure of 1 indicates closeness similarity and 0 means no similarity at all.

Hashtag 1	Hashtag 2	Levenshtein	Jaro	Jaro-Winkler
#20delecciones	*#20democracia*	0.3846	0.6773	0.8064
#20delecciones	*#20dediciembre*	0.3846	0.7019	0.8211
#7deldebatedecisivo	*#7deldebateadecisivo*	0.9473	0.9824	1.0000
#7deldebatedecisivo	*#7deldebatedecsivo*	0.9445	0.9618	1.0000
#canarias	*#valencia*	0.2500	0.5834	0.5834
#marianorajoy	*#pedrosanchez*	0.0834	0.3889	0.3889

Table 3: String metrics between hashtags for different examples of dataset *Elecciones*

In order to unify similar hashtags the first approach could be to calculate distances between all hashtags of a dataset. However this process implies a quadratic complexity on the number of hashtags. Concretely, if we have n hashtags, we need to compute $\frac{n(n-1)}{2}$ pairwise distances. For instance, given the *Elecciones* dataset, with 7950 unique hashtags, we would need to compute 31597275 string distances. Due to its large complexity, this complete method is not feasible for medium size datasets. As a result, we propose in this paper a filter to group similar hashtags based on the alphabetical order:

1. We sort the n hashtags list in alphabetical order
2. We calculate the distance between one hashtag and the nearest k neighbours in the list.
3. Given a level of similarity α, starting from the beginning and in alphabetical order, we group hashtags with a similarity more or equal than α.

Note that using alphabetical order and computing distances between neighbours we only need n pairwise distance computations. This approximation has important limitations. For instance, if the spelling error is located in the first characters, the algorithm will not group properly this hashtag. We can also improve the performance of the filter using more than one neighbour (factor k) in the step 2 and 3, but this also could increase the time complexity of the filter. This k factor could be established depending on the size of the dataset. In this work we only consider the nearest neighbour, $k = 1$.

4 Experiments

After the correct identification of hashtags, in this section we study the distribution of hashtags for both datasets. In particular we analyse if the frequency distribution of hashtags follow Benford's and Zipf's law.

4.1 Zipf's law

First, we compare the frequency distribution of hashtags with respect to Zipf's law.

4.1.1 Dataset *Hispatweets*

If we analyse separately the frequency distribution of hashtags for each one of the countries of the dataset *Hispatweets*, we observe that all of them present a close distribution with respect to Zipf's law. Table 4 includes the regression line (considering a log-log scale) induced for the frequency distribution and the coefficient of determination R^2 computed with respect to Zipf's law distribution. Since all values are close to -1, we can see that the frequency distribution of hashtags follow approximately Zipf's law. Figure 2 shows an example of the line induced by regression with respect to the ideal Zipf's law.

[5]https://pypi.python.org/pypi/python-Levenshtein/0.12.0

Country	Regression line	R^2
Argentina	$-1.1011x + 4.4794$	-0.9549
Chile	$-0.9538x + 4.4206$	-0.9617
Colombia	$-0.9550x + 4.3778$	-0.9641
Spain	$-0.9496x + 4.5036$	-0.9628
Mexico	$-0.8612x + 4.0208$	-0.9527
Peru	$-0.8953x + 4.1562$	-0.9549
Venezuela	$-1.0394x + 4.8159$	-0.9617

Table 4: Regression lines induced from frequency of hashtags for each country of *Hispatweets* dataset. Coefficient of determination R^2 computed with respect to Zipf's law distribution

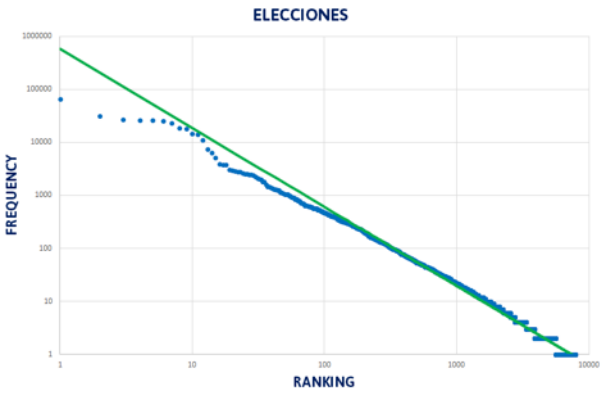


Figure 2: Regression lines induced from frequency of hashtags for Spain with respect to Zipf's law distribution (considering a log-log scale).

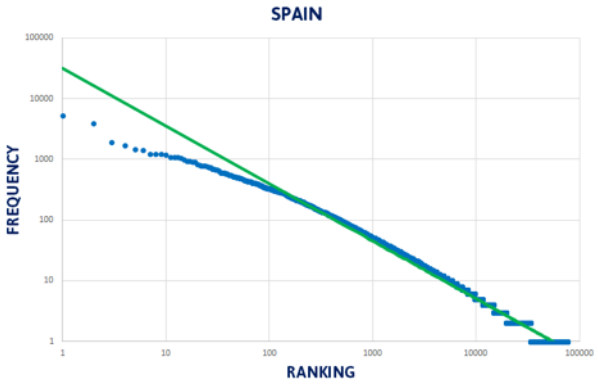


Figure 3: Regression lines induced from frequency of hashtags for dataset *Elecciones* with respect to Zipf's law distribution (considering a log-log scale).

4.1.2 Dataset *Elecciones*

For this dataset the distribution of the frequency of hashtags is again very close to Zipf's law distribution. Using a log-log scale, the distribution is approached by linear regression to a the following line: $-1.4909x + 5.7644$. Here, the Coefficient of determination $R^2 = -0.9879$ is extremely close to -1. Figure 3 includes the line induced by regression for this dataset with respect to the ideal Zipf's law.

4.2 Benford's law

After analysing the Zipf's law on the two datasets with succesful results, here we study if the distributions of the frequency of hashtags follow Benford's law.

4.2.1 Dataset *Hispatweets*

Table 5 shows the percentage of each FSD (*First Significant Digit*) for the seven countries of the dataset. We also include in the first row the theoretical percentage for each FSD according to the Benford's law. We can observe that, for all cases, there are important differences between the computed FSD values and the theoretical values expected by Benford's law. The disparity is specially great for the case FSD = 1, mainly because we have detected a gross number of hashtags that only appear once. In part, this is caused because sometimes Twitter users introduce unintended mistakes when writing hashtags, and then, they are accounted as different. In order to correct these wrong hashtags we try to unify some of them according to the procedure explained in

Section 3. We have tested three edition distances: Levenshtein, Jaro and Jaro-Winkler. In short, Levenshtein distance counts the number of editions (insertions, deletions, or substitutions) needed to convert one string into the other. Jaro gives a measure of characters in common, being no more than half the length of the longer string in distance, with consideration for transpositions. The modification included in Jaro-Winkler takes the idea that differences near the start of the string are more significant than differences near the end of the string, see for instance (Naumann and Herschel, 2010). All of them range from 0 to 1, with 1 representing the case of coincidence.

According to our results, this last distance is the most valid to unify similar hashtags. In Table 6 we include the values of the FSD for the case of Spain and different values of α. According to these results, $\alpha = 0.8$ is the value that obtains better results when we compare the distribution of FSD with respect to the FSD according of the Benford's law. Similar results have been obtained for the rest of countries.

4.2.2 Dataset *Elecciones*

We also have a similar result in the case of dataset *Elecciones*. Table 7 includes the computed distribution of FSD without filtering hashtags, and applying the filter based on Jaro-Winkler distance for different values of α. Again, we find a situation with a high number of hashtags with just one appearance. After applying the filter, we reduce this situation by joining hashtags that probably were different because of type-writing errors. As in the previous dataset, $\alpha = 0.8$ is the value that obtains more similar results to the theoretical estimates of FSD according to Benford's law.

4.3 Analysis of results

According to the results presented in the analysed datasets, we can observe that when we study a significant number of tweets, the distribution of the FSD approaches to Benford's law, specially if we apply a filter step that joins similar hashtags. In order to assess this conclusion, we introduce in this part some experiments where we measure the similarity between the computed distribution of FSDs with respect to the theoretical expected FSD distribution defined by Benford's law.

In Table 9 we include some measures for evaluating the similarity between the computed and theoretical distribution of FSDs. These are:

- **Pearson Correlation**: a measure for estimating the linear dependence between two variables. The estimated value is between +1 (total positive linear correlation) and -1 (total negative linear correlation). Correlation 0 indicates no linear correlation.

- χ^2: This metric is defined as the difference of the computed distribution with respect to the theoretical distribution:

$$\chi^2 = \sum_{d=m}^{9} \frac{(P_{obs}(d) - P_t(d))^2}{P_t(d)} \qquad (3)$$

  where:

  - $P_t(d)$ is the theoretical frequency and $P_{obs}(d)$ is the observed frequency
  - m refers to the analysed digit. Here we study the first digit, thus $m = 1$.

  Since χ^2 estimates the difference between distributions, lower values of the metric indicates distributions closer to Benford's law. According to (Nigrini, 2012), we can assume that a distribution does not follow Benford's law for the first digit (FSD) if $\chi^2 > 15.507$ (confidence 95%), and if $\chi^2 > 20.090$ (confidence 99%).

- **Mean absolute deviation (MAD)** : The average absolute deviation (or mean absolute deviation) is a summary statistic of dispersion. MAD estimates the average of the absolute deviations from a theoretical distribution. For Benford's law, it is computed in the following way:

$$MAD = \frac{1}{9} \sum_{d=1}^{9} |P_{obs}(d) - P_t(d)| \qquad (4)$$

  For making a hypothesis contrat, we consider as null hypothesis that a distribution follows Benford's law. Since χ^2 estimates the difference between distributions, lower values of the metric indicates distributions closer to Benford's law. According to (Nigrini, 2012), we use this metric to estimate different values of conformity of a distribution with respect to Benford's law. These ranges are presented in Table 8.

	1	2	3	4	5	6	7	8	9	Total
FSD Benford	30.01%	17.60%	12.40%	9.69%	7.91%	6.69%	5.79%	5.11%	4.57%	100%
FSD Argentina	62.54%	13.37%	6.69%	4.14%	2.41%	1.80%	1.18%	1.03%	0.83%	100%
FSD Chile	60.88%	19.19%	6.95%	4.51%	2.86%	2.11%	1.47%	1.16%	0.87%	100%
FSD Colombia	59.45%	19.41%	7.36%	4.71%	3.05%	2.30%	1.49%	1.28%	0.95%	100%
FSD Spain	60.17%	19.89%	7.00%	4.56%	2.74%	2.03%	1.52%	1.16%	0.92%	100%
FSD Mexico	63.53%	18.61%	6.41%	3.98%	2.52%	1.84%	1.33%	0.96%	0.82%	100%
FSD Peru	62.60%	19.15%	6.78%	4.08%	2.47%	1.84%	1.23%	1.05%	0.79%	100%
FSD Venezuela	58.71%	19.55%	7.48%	4.89%	3.01%	2.01%	1.61%	1.32%	1.12%	100%

Table 5: Percentage of each FSD *First Significant Digit* for the seven countries of the dataset *Hispatweets*. The first row contains the expected Percentage of each FSD according to Benford's law.

	Jaro-Winkler Distance								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
FSD Benford	30.01%	17.60%	12.40%	9.69%	7.91%	6.69%	5.79%	5.11%	4.57%
FSD Spain	60.17%	19.89%	7.00%	4.56%	2.74%	2.03%	1.52%	1.16%	0.92%
$\alpha = 0.95$	54.57%	20.48%	8.46%	5.58%	3.60%	2.61%	1.96%	1.47%	1.27%
$\alpha = 0.90$	49.78%	20.70%	9.48%	6.55%	4.40%	3.13%	2.41%	1.92%	1.62%
$\alpha = 0.85$	44.74%	20.51%	10.62%	7.18%	5.25%	4.02%	2.95%	2.67%	2.06%
$\alpha = 0.80$	39.89%	20.56%	11.12%	8.33%	5.92%	4.72%	3.45%	3.49%	2.55%

Table 6: Percentage of each FSD for Spain in dataset *Hispatweets* applying a filter based on Jaro-Winkler distance and different values of α. The first row contains the expected percentage of each FSD according to Benford's law. The second row contains the percentage of each FSD without hashtag the union filter.

	Jaro-Winkler Distance								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
FSD Benford	30.01%	17.60%	12.40%	9.69%	7.91%	6.69%	5.79%	5.11%	4.57%
FSD *Elecc.*	40.39%	25.99%	8.97%	9.23%	3.91%	4.50%	2.54%	2.69%	1.77%
$\alpha = 0.95$	39.00%	24.61%	9.85%	10.07%	4.45%	4.88%	2.51%	2.79%	1.83%
$\alpha = 0.90$	38.08%	23.24%	10.60%	9.79%	4.92%	5.09%	3.24%	3.07%	1.98%
$\alpha = 0.85$	36.35%	21.88%	11.55%	9.80%	5.42%	5.51%	3.61%	3.49%	2.40%
$\alpha = 0.80$	33.8%	20.49%	12.56%	9.70%	6.61%	5.81%	4.13%	4.27%	2.63%

Table 7: Percentage of each FSD for dataset *Elecciones* applying a filter based on Jaro-Winkler distance and different values of α. The first row contains the expected percentage of each FSD according to Benford's law. The second row contains the percentage of each FSD without the hashtag union filter.

Range	Conformity Level
0.000 to 0.006	High
0.006 to 0.012	Good
0.012 to 0.015	Medium
0.015 or more	Low

Table 8: Range of critical values and the corresponding conformity level for Mean absolute deviation and Benford's law on the first significant digit.

Distribution	Correlation	χ^2	MAD
Spain	0.9699	51.41	0.071
Spain + J-W $\alpha = 0.8$	0.9966	7.48	0.028
Elecciones	0.9835	23.78	0.038
***Elecc.* + J-W** $\alpha = 0.8$	0.9979	2.72	0.014

Table 9: Pearson Correlation, χ^2 statistics and Mean absolute deviation (MAD) between observed distribution of FSD and theoretical distribution of FSD according to Benford's law. We include original datasets and datasets after applying the Jaro-Winkler Distance filter.

If we analyse the results of Table 9, we can observe that for all cases correlation obtain high values (greater than 0.92). We can see that corrected versions for both datasets increase the correlation with respect to Benford's law.

A similar behaviour is observed in χ^2 statistics. The Jaro-Winkler Distance filter is able to unify numerous hashtags and then the similarity with respect to to Benford's law is drastically increased. If we consider the test proposed by (Nigrini, 2012), and the corrected version of the dataset, the hypothesis that distributions does not follow Benford's law cannot be rejected.

Finally, considering Mean absolute deviation (MAD), we find the same pattern. Jaro-Winkler Distance filter reduces the distance between distributions. In this case, the test proposed by (Nigrini, 2012) determines that *Spain* dataset has a low similarity with respect to Benford's law, and the *Elecciones* dataset (corrected version) has a medium similarity. These results are in some cases contradictory with respect to the conclusions observed with χ^2 statistics, and indicate that MAD test seems to be more strict that χ^2 test.

5 Conclusions

Benford's Law is useful to estimate the probabilities of highly likely or highly unlikely frequencies of numbers in datasets. Those who are not aware of this experimental law and intentionally manipulate numbers are susceptible to be discovered by the comparison with respect to Benford's Law. We find examples of this use in electoral processes, accounting fraud detection, scientific fraud detection...

In this paper, Benford's and Zipf's laws have been testing against hashtag frequency on datasets of tweets. A similar analysis has been recently checked for the case of followers distributions in Facebook, Twitter (Golbeck, 2015). We confirm that the distribution of hashtag frequency follows a power law, as Zipf's law expects. That is, few hashtags achieve a high number of mentions, and most of them lack of impact with few repetitions . The source of this dispersion is probably the lack of control of Twitter on the use of hashtags. The social network permits that hashtags can be created without any restriction, and it also lacks of a recommender system for the generation of hashtags. In fact, we detected an irregular number of hashtags with just one mention. Many of these hashtags are spelling mistakes of Twitter users. In order to mitigate this dispersion, we defined a union filter based on string distances that is able to group filters based on their similarity. We use alphabetical order of hashtags in order to reduce time complexity of the cluster algorithm. The comparison of three string distances *Levenshtein*, *Jaro* and *Jaro-Winkler* indicates that the last one, *Jaro-Winkler*, obtains the better performance in correcting hashtags.

We also analyse the distribution of the first significant digit of the hashtag frequencies with respect to Benford's law. Experiments on the datasets of tweets considering three different metrics: Pearson Correlation, χ^2 and Mean absolute deviation, reveal that this law is approximately followed by the distribution of the first significant digit of the hashtag frequencies, specially when we apply a group filter based on the *Jaro-Winkler* distance in order to correct spelling errors in hashtags. In order to give statistical significance to our research, we apply some of the tests provided by (Nigrini, 2012) that allow to verify the level of conformity of a frequency distribution with respect to Benford's law. According to the results,

χ^2 test returns high level of conformity, while considering Mean absolute deviation (MAD), we get medium and low level of conformity. These two tests are in some way contradictory and show that MAD test seems to be more strict that χ^2 test.

As future work, we propose the improvement of the hashtag unification filter by improving the mechanism for detecting similarities between hashtags. We will also study the applicability of the experimental laws on bigger tweet datasets, where, likely, the levels of conformity will be greater.

Acknowledgments

We thank Francisco Almenar Pedrós, José Francisco García Cantos, and Mirella Oreto Martínez Murillo for providing us the dataset *Elecciones*. We also thank Raül Fabra Boluda, Paolo Rosso, and Francisco Manuel Rangel Pardo for providing us the dataset *Hispatweets*. This work has been partially supported by the EU (FEDER) and Spanish MINECO grant TIN2015-69175-C4-1-R, LOBASS and the REFRAME project, granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences Technologies ERA-Net (CHIST-ERA), and funded by MINECO in Spain (PCIN-2013-037) and by Generalitat Valenciana PROMETEOII/2015/013.

References

Stephen Battersby. 2009. Statistics hint at fraud in iranian election. *New Scientist*, 24.

Frank Benford. 1938. The law of anomalous numbers. *Proceedings of the American Philosophical Society*, pages 551–572.

Dorina Bërdufi. 2013. Statistical detection of vote count fraud: 2009 albanian parliamentary election and Benford's law. *Academic Journal of Interdisciplinary Studies*, 2(8):379.

Hsia-Ching Chang. 2010. A new perspective on twitter hashtag use: Diffusion of innovation theory. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *International Joint Conference on Artificial Intelligence (IJCAI) 18, Workshop on Information Integration on the Web*.

Cindy Durtschi, William Hillison, and Carl Pacini. 2004. The effective use of Benford's Law to assist in detecting fraud in accounting data. *Journal of forensic accounting*, 5(1):17–34.

Miles Efron. 2010. Hashtag retrieval in a microblogging environment. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788. ACM.

Raül Fabra-Boluda. 2016. Identificación de la variedad del lenguaje para la mejora del geoposicionamiento en social media. Master's thesis, Universitat Politècnica de València. `http://users.dsic.upv.es/~prosso/resources/FabraMSc.pdf`.

James L. Friar, Terrance Goldman, and Juan Pérez-Mercader. 2012. Genome sizes and the Benford distribution. *PLoS One*, 7(5):e36624.

Jennifer Golbeck. 2015. Benford's Law Applies to Online Social Networks. *PLOS ONE*, 10(8):e0135169.

Matthew J. Hickman and Stephen K. Rice. 2010. Digital analysis of crime statistics: Does crime conform to Benford's law? *Journal of Quantitative Criminology*, 26(3):333–349.

Shi-Ming Huang, David C. Yen, Luen-Wei Yang, and Jing-Shiuan Hua. 2008. An investigation of Zipf's law for fraud detection. *Decision Support Systems*, 46(1):70–83.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193. Association for Computational Linguistics.

David Burth Kurka, Alan Godoy, and Fernando J. Von Zuben. 2016. Online social network analysis: A survey of research applications in Computer Science. *Preprint*, page arXiv:1504.05655.

Zongyang Ma, Aixin Sun, and Gao Cong. 2012. Will this # hashtag be popular tomorrow? In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1173–1174. ACM.

Hans Christian Müller. 2011. Greece was lying about its budget numbers. *Forbes*, 12.

Felix Naumann and Melanie Herschel. 2010. An introduction to duplicate detection. *Synthesis Lectures on Data Management*, 2(1):1–87.

Mark Nigrini. 2012. *Benford's Law: Applications for forensic accounting, auditing, and fraud detection*, volume 586. John Wiley & Sons, Hoboken, NJ.

Ioan-Iovitz Popescu. 2003. On a Zipf's law extension to impact factors. *Glottometrics*, 6:83–93.

David M. W. Powers. 1998. Applications and explanations of Zipf's law. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pages 151–160. Association for Computational Linguistics.

Sriram Srinivasan, Sourangshu Bhattacharya, and Rudrasis Chakraborty. Segmenting web-domains and hashtags using length specific models. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*.

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, Cambridge, MA.

A Multi-aspect Analysis of Automatic Essay Scoring for Brazilian Portuguese

Evelin Carvalho Freire de Amorim
Computer Science Department
Universidade Federal de Minas Gerais
Minas Gerais, Brazil
evelin.amorim@dcc.ufmg.br

Adriano Veloso
Computer Science Department
Universidade Federal de Minas Gerais
Minas Gerais, Brazil
adrianov@dcc.ufmg.br

Abstract

While several methods for automatic essay scoring (AES) for the English language have been proposed, systems for other languages are unusual. To this end, we propose in this paper a multi-aspect AES system for Brazilian Portuguese which we apply to a collection of essays, which human experts evaluated according to the five aspects defined by the Brazilian Government for the National High School Exam (ENEM). These aspects are skills that student must master and every skill is assessed separately from one another.

In addition to prediction, we also performed feature analysis for each aspect. The proposed AES system employs several features already used by AES systems for the English language. Our results show that predictions for some aspects performed well with the employed features, while predictions for other aspects performed poorly.

Furthermore, the detailed feature analysis we performed made it possible to note their independent impacts on each of the five aspects. Finally, aside from these contributions, our work reveals some challenges and directions for future research, related, for instance, to the fact that the ENEM has over eight million yearly enrollments.

1 Introduction

The goal of automatic essay scoring (AES) systems is to score a given essay. AES systems are relevant for educational institutions, since the human effort to evaluate essays is high and, and students need feedback to improve his or her writing skills. Besides these issues, almost every senior high school student in Brazil should write an essay to the National Exam of High School (ENEM), which Brazilian government uses to evaluate the quality of high schooler's education and that of their institution.

Although there are thousands of essays written to ENEM every year, to the best of our knowledge there is no AES system for Brazilian Portuguese (BP) language, or an analysis of features in a multi-aspect essay scoring system for BP. Each aspect is a skill that students must master as seniors in high school. Nonetheless, several AES systems have been proposed for the English Language. Attali and Burstein (Attali and Burstein, 2006) proposed an AES system, called e-rater, that employs general features of argumentative essays to scoring prediction. The main features used by e-rater are grouped into the following types: grammar, usage, mechanics, and style; organization and development; lexical complexity; and prompt-specific vocabulary usage. e-rater employs multiple regression, and it is task independent AES system, i.e., its score is independent of the given prompt.

Napoles and Callison-Burch (Napoles and Callison-Burch, 2015) employ linear regression to an AES system that intends to assign more uniform grades than multiple human evaluators. Similar to our task, Napoles and Callison-Burch propose the task of multi-aspect classification using five grading categories. However, the authors leave unexplained how each of their aspects is affected by their features. We think this is a significant contribution since students or professors can use features as a feedback for better understanding essays writing. Besides that, Napoles and Callison-Burch assume that more than one evaluator is available to train their model, which in the real world is not always the case.

Larkey (Larkey, 1998) proposed three models

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 94–102, Valencia, Spain, April 3-7 2017.

that are based on text classification to score essays applying linear regression. However, Larkey strategy is task-dependent. Chen and He (Chen and He, 2013) also grouped features into four main types: lexical features; syntactical features; grammar and fluency features; content and prompt-specific features. Then, the authors proposed a rank-based algorithm that maximizes the agreement between human score and machine score. Zesh et al. (Zesch et al., 2015) developed a technique that adapts domain in an AES system. The authors tested their method in an English dataset and a German dataset. Chali and Hasan (Hasan, 2012) proposed an LSA-based method, that is task-dependent, and the goal was to establish a strategy to understand the inner meaning of texts. Beyond the English language, Kakkonen and Sutinen (Kakkonen and Sutinen, 2004) developed an AES system to the Finnish language also based on LSA algorithm.

Besides assigning grade score, other researches proposed to analyze argumentation strength of essays (Persing and Ng, 2015), discourse structure of essays (Song et al., 2015)(Stab and Gurevych, 2014) , and grammar correction in general (Rozovskaya and Roth, 2014)(Lee et al., 2014).

Our research is different from the previous research since we aim to answer the following questions:

1. How objective features behave in a multi-aspect automatic essay scoring system?
2. Which features are more relevant for each aspect?

In addition to this, we aim to pose some interesting questions for future research. Our essays present not only grades but also evaluator's comments about the aspects that are considered in the ENEM. During the exploration of evaluators comments, bias was observed in some evaluations, which we define as being when some human evaluator seems to disagree or agree with student's point of view, which can lead to an improper influence on the student's grade. The possibility of bias of human evaluations raises some questions.

1. Are some topics for essays more prone to result in biased evaluation?
2. Is it possible to detect if human evaluator is biased for or against a given student's point of view?
3. If it is possible to detect the bias of human evaluator, is it feasible to measure the quantitative affect on grades?
4. Is there any difference between the words in biased evaluations that agrees with student point of view and biased evaluations that disagrees with student's point of view?

In a nutshell, the availability of evaluator comments allows for a host of issues related to bias detection, quantification, and resolution, yet as far as we know these questions are still unanswered.

The paper is organized as follows. The second section details our dataset and the features we use. The third section explains the experiments we performed and the results of our experiments. The fourth section presents the main remarks about our research and the fifth section point to the future direction for our research.

2 Methodology

We propose a methodology that besides the usual features employed by popular AES methodologies (Attali and Burstein, 2006) (Chen and He, 2013) (Zesch et al., 2015), it also takes advantage of domain features. To test our proposed features, we used a dataset of nearly 1840 essays. Next sections describe our dataset and our features.

2.1 Dataset

Our dataset is composed of 1840 essays about 96 topics, which were crawled from UOL Essay Database website[1]. The average length in words are 300.51; the biggest essay has 1293 words, and the smallest essay has 49 words. Each essay is evaluated according to the following five aspects:

1. **Formal language**: Mastering of the formal Portuguese language.
2. **Understanding the task**: Understanding of essay prompt and application of concepts from different knowledge fields, to develop the theme in an argumentative dissertation format.
3. **Organization of information**: Selecting, connecting, organizing, and interpreting information, facts, opinions, arguments to advocate a point of view.

[1] http://educacao.uol.com.br/bancoderedacoes

Table 1: Score and corresponding levels

Score	Level
2.0	Satisfactory
1.5	Good
1.0	Regular
0.5	Weak
0.0	Unsatisfying

Table 2: Average Score for each aspect and final grade in UOL Dataset

Aspect	Average Score
Formal Language	1.1
Understanding the task	0.91
Organization of information	0.93
Knowing argumentation	0.83
Solution proposal	1.08
Final grade	4.86

4. **Knowing argumentation**: Demonstration of knowledge of linguistic mechanisms required to construct arguments.

5. **Solution proposal**: Formulation of a proposal to the problem presented, respecting human rights and considering socio-cultural diversity.

Each aspect is scored according to the scale of Table 1, and the final score is the sum of all aspects scores. Table 2 depicts the average score assign by humans for each aspect and final grade in our dataset.

Each essay is evaluated by only one human. Although this seems a disadvantage, we think that this is a real world dataset, since in most high schools only one teacher scores essay. Also, as we aim to detect the impact of features in each aspect, one evaluator per essay is enough.

2.2 Features

Features are divided into two main types, domain features that are related to ENEM exam or Brazilian Portuguese Language, and general features that are based on Attali and Burstein research (Attali and Burstein, 2006).

1. *Domain features:* ENEM exam doesn't allow the using of the first person pronouns and verbs. Therefore, we employ as features the number of first person pronouns and verbs and the number of first person pronouns and verbs per number of tokens. Also, we suggest as feature the number of *ênclise*, a Portuguese language structure, and the number of *ênclise* per number of tokens. *Ênclise* is unusual to BP spoken language, then if a student applies such concept in essay, probably he or she knows how to use formal language better. Also, the excessive number of demonstrative pronouns is condemned in written BP (Martins, 2000); then we use the number of demonstrative pronouns and the number of demonstrative pronouns per number of tokens.

2. *General:* Most of the general features are based on Attali and Burstein (Attali and Burstein, 2006) research, which presented ten features. However, due to lack of tools for Brazilian Portuguese and time constraints, we implemented only six features and adapted two features from the e-rater framework. Next, we detailed our feature implementation.

   - *Grammar and style:* Grammar was checked by CoGrOO (Kinoshita et al., 2006), which is a Brazilian add-on to Open Office Writer. Also, for spelling mistakes we use a Brazilian software [2]. Both features were also divided by the number of tokens in an essay; then we employed four features for grammar and spelling errors. To evaluate style in essays, we applied LanguageTool rules for Portuguese, but also we added some rules suggested by a Portuguese manual of writing (Martins, 2000)[3]. We employed the number of style errors and the number of style of errors per sentence as features.
   - *Syntactical features:* According to (Martins, 2000), in Portuguese Language, sentences longer than 70 characters are long sentences, and therefore are not recommended. We employ as a feature, the number of sentences longer than 70 characters.
   - *Organization and development:* There

[2] https://github.com/giullianomorroni/JCorretorOrtografico

[3] Rules can be examined in `https://goo.gl/F32hcC`

are no tools to evaluate organization and development in Portuguese language, then we collected discourse markers in a Brazilian Portuguese grammar (Jubran and Koch, 2006). Discourse markers are linguistic units that establish connections between sentences to build coherent and knit discourse. We employed as features the number of discourse markers and the number of discourse markers per sentence.

- *Lexical complexity:* To evaluate lexical complexity, we used four features. The first feature is Portuguese version of Flesh score (Martins et al., 1996); the second feature is average word length, which length is the number of syllables; the third feature is the number of tokens in an essay; the fourth feature is the number of different words in an essay.
- *Prompt-specific vocabulary usage:* It is desirable to employ concepts from the prompt in the essay, therefore for each essay we compute cosine similarity between prompt and essay. In this case, the prompt is a frequency vector of words, and the essay is also a frequency vector of words, which are from the prompt vocabulary. We decided for this strategy since, unlike other works, our dataset comprises many different topics, each with few essays. Then, we think that build a vocabulary for each domain it is not helpful.

3 Experiments

We performed two types of experiments: one evaluating the performance of grade prediction for each aspect and other evaluating the role of each feature in grade prediction task. Feature analysis is of particular importance for this task since computer evaluation of an essay is different from a human analysis. Therefore, explore which variable is important for which aspect is crucial for the development of our research.

3.1 Prediction Analysis

Besides ASAP challenge at Kaggle[4], several works employ **quadratic weighted kappa** as the evaluation metric (Zesch et al., 2015)(Chen and He, 2013)(Attali and Burstein, 2006), which aims to measure agreement between human evaluation and machine scoring. When the value of kappa is closer to 1, the higher the agreement between evaluators, and when the value of kappa is closer to 0, the lower the agreement between evaluators.

[4]https://www.kaggle.com/c/asap-aes/details/evaluation

Table 3: List of Features grouped into domain type and general type

Group	Feature
Domain	#first person of singular of verbs and pronouns
	#first person of singular of verbs and pronouns / #tokens
	#demonstrative pronouns
	#demonstrative pronouns / #tokens
	#enclise
	#enclise / #tokens
General	#sentences longer than 70 characters
	#grammar errors
	#grammar errors / #token
	#spelling errors
	#spelling errors / #token
	#style errors / #sentences
	#discourse markers
	#discourse markers / #sentence
	Flesh score
	Average word length (syllables)
	#tokens
	similarity with prompt
	#different words

First, we compute a matrix of weights (Equation 1) that are based on the difference between human evaluation and machine scoring.

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \quad (1)$$

The second step calculates a histogram matrix called O, where $O_{i,j}$ is the number of essays that receive grade $i \in N$ by a human evaluator and a grade $j \in N$ by a machine evaluator. After that, we built another matrix E of expected ratings, which is the outer product between each rater's histogram vector of ratings. Finally, we employ O, E, and w to compute the quadratic weighted kappa using Equation 2.

Table 4: Kappa values for each grade aspect

Grade Type	Kappa
Final Grade	0.3673
Formal Language	0.3147
Understanding the task	0.2678
Organization of Information	0.2305
Knowing argumentation	0.2704
Solution proposal	0.1393

Table 5: Kappa values for each grade aspect after oversampling (full and general feature set)

Grade Type	Full	General
Final Grade	0.4245	0.4131
Formal Language	0.3351	0.3249
Understanding the task	0.1817	0.1822
Organization of Information	0.2728	0.2679
Knowing argumentation	0.2668	0.2484
Solution proposal	0.1542	0.1430

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \quad (2)$$

A simple regression is applied to predict the final grade of essays, and each of other five aspects. Also, a simple oversampling strategy is applied since grade distribution is unbalanced (Figure 1).

In the first step of oversampling strategy, it searches by the class G_{max} that holds the largest number of instances. Then the strategy randomly selects instances from every class $G \neq G_{max}$ and replicates such instances into training datasets, until the size of every class $G \neq G_{max}$ be equal the size of G_{max}.

Table 4 describes the results using quadratic weighted kappa before the oversampling. We executed cross-validation five times and compute the average of kappas of all experiments, for each aspect and final grade, to evaluate oversampling performance. Results after oversampling are described in Table 5.

Considering the lack of tools for processing the Portuguese language, and the limited performance of the few existing tools, the multi-aspect classification performed satisfactorily. However, some aspects performed poorly probably due to the subjectivity intrinsic to these aspects and objective variables probably can't capture all the subjectivity.

3.2 Feature Analysis

Besides kappa results, we also performed an experiment that investigates the impact of each feature in each aspect and final grade. The experiments were performed removing each feature and measuring the resulting kappa. If removing a feature f lowers the resulting kappa, then that feature is relevant to the model of that aspect. According to this criterion, the lower the resulting kappa when removing f from the training model, the more important is f for this model. Table 3.2 describes the three features that most diminished kappa value and the three features that most increased kappa value. The **full** value in table present the result with the full set of features described earlier.

It is possible to observe that the most relevant features for the final grade are not necessarily a mix of relevant features from the aspects. For instance, vocabulary level is one of three most important features for the final grade, but, while not irrelevant, it is not in the top three for the aspects. To understand better the role of vocabulary level, we compute in our dataset average vocabulary level for the final grade, and, as expected, the higher the grade, the higher the number of different words in essays. Besides vocabulary level, lexical complexity seems to play a significant role to final grade, since three of the most important features to final grade prediction affect prediction.

Aspect *Understanding the task* presented the lowest kappa value between aspects. However, we can draw some conclusions from Table 3.2. For instance, Flesh score affected expressively kappa value. Also, we observe that current features are not enough for *Understanding the task* model, therefore we will implement new features related to this aspect.

Organization of information resulted in the second highest kappa value between aspects. As *similarity to prompt* was the most relevant features, we believe that similarity between semantic vectors, as proposed by Zesh et. al (Zesch et al., 2015), also can improve *Organization of Information* prediction. Another observation is the influence of style errors upon *Organization of Information* aspect. Perhaps this influence is because the definition of style we used is related to how the writer "present" the information, which can be redundancies or nonexistent language expressions.

With respect to the *Knowing argumentation* as-

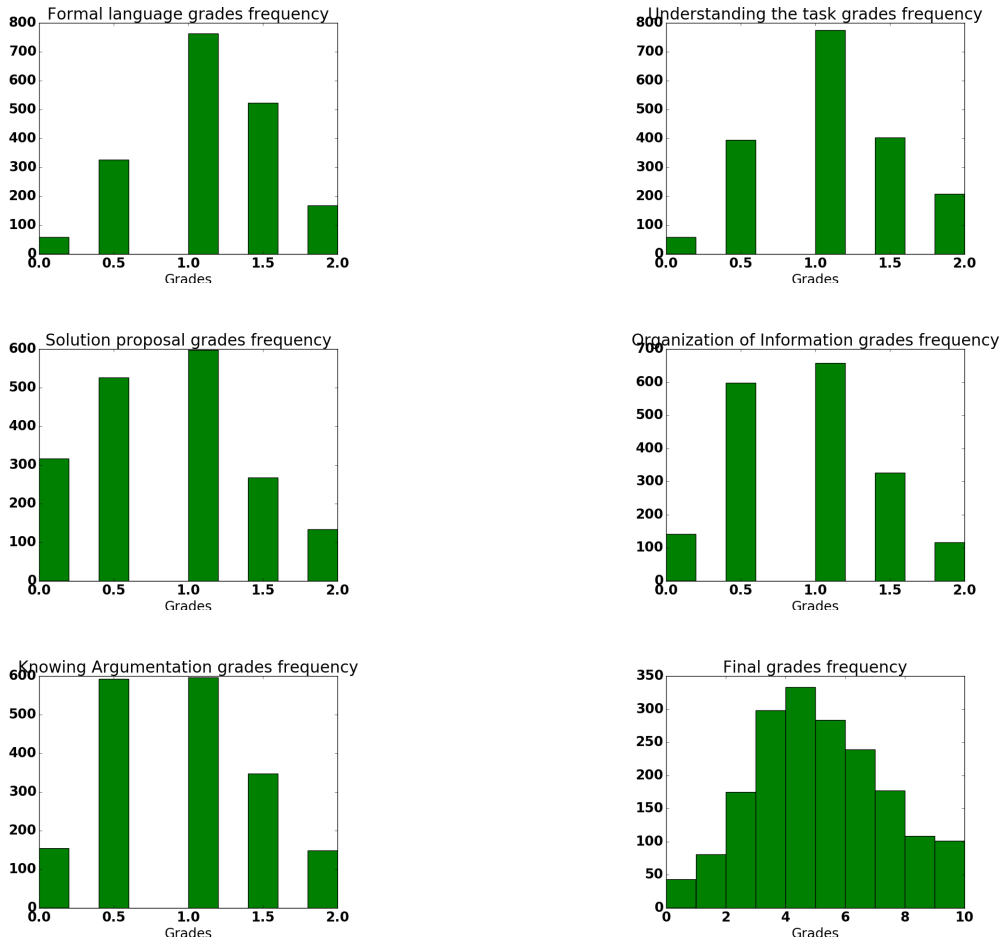


Figure 1: Distribution of grades in UOL dataset for each aspect and final grade

pect, we believe that style errors affected results for a similar reason that we mentioned in the analysis of *Organization of information* aspect. However, in regard this aspect we think that perhaps some argument features ((Stab and Gurevych, 2014), (Song et al., 2015)) will improve *Knowing argumentation* scoring prediction.

4 Conclusion

We proposed a multi-aspect automatic essay correction system for Brazilian Portuguese. Our primary goal is to evaluate if classical features for AES system for the English language performs well in a multi-aspect scenario, and assess which features are important for which aspect. In fact, after experiments, some features performed well for some aspects. Nonetheless, each aspect performed in a different way, which suggests that each aspect needs an own suitable model. Also, more specific features for some aspects probably will enhance subjective aspects.

Academic level, represented by Flesh score, is extremely relevant in most aspects. A possible reason for these results is because a high school student should present advanced skills, like grammar, spelling, argumentation, among others. Despite this feature in common, each aspect exhibits their singularity. Like enclise affecting *Understanding the task*, similarity with prompt influencing *Organization of information*, and discourse markers changing *Solution proposal*. Therefore, while some of the features enhance results for some aspects, these same features harm prediction for other aspects.

5 Future Directions

The following issues are directions we aim to pursue in our further research.

Analysis of evaluators comments. Our dataset comprises human evaluators comments. We intend to analyze these comments, which is of particular importance for argumentative essays since the opinion of human evaluators about a topic can affect grades. In a sample of 48 essays taken from our dataset, two linguists detected that 11 essays presented biased evaluation. Biased evaluation is a more serious issue if we will think about ENEM and other tests that are a relevant factor to many students. Some works were performed in bias language, but none of them analyzed bias on evaluations. Also, we can apply the same reasoning for other types of evaluations, like peer review of papers. Besides that, we would like to research how we can minimize bias on automatic scoring prediction.

Composite Classifier. A classifier to predict final grades employing predictions of the five aspects is a natural step in our research.

Adding new features to Brazilian Portuguese AES. There are more features to add to Brazilian Portuguese AES. Some of these features are: POS-tagging ratio; word length in characters; the number of commas, quotations or exclamation marks; average sentence length; average depth of syntactic trees; and topical overlap between adjacent sentences. Also, cohesion features like proposed by Song et al. (Song et al., 2015) can improve aspects like *Solution Proposal*, which probably demands sophisticated features.

6 Acknowledgements

We would like to thank Marcia and Luana, the two linguists that have been assisting us on bias research.

We thank the partial support given by the Brazilian National Institute of Science and Technology for the Web (grant MCT-CNPq 573871/2008-6), Project Models, Algorithms and Systems for the Web (grant FAPEMIG/PRONEX/MASWeb APQ-01400-14), and authors individual grants and scholarships from CNPq and CAPES.

References

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, Seattle, WA, USA.

Yllias Chali Sadid A Hasan. 2012. Automatically assessing free texts. In *24th International Conference on Computational Linguistics*, page 9, Bombay, India.

Clélia Jubran and Ingedore Koch. 2006. *Gramática do português culto falado no Brasil: construção do texto falado*, volume 1. UNICAMP.

Tuomo Kakkonen and Erkki Sutinen. 2004. Automatic assessment of the content of essays based on course materials. In *2nd International Conference on Information Technology: Research and Education*, pages 126–130, Semarang, Indonesia. IEEE.

Jorge Kinoshita, Lais N. Salvador, and Carlos E. D. Menezes. 2006. Cogroo: a brazilian-portuguese grammar checker based on cetenfolha. In *The fifth international conference on Language Resources and Evaluation (LREC)*, pages 2190–2193, Genova, Italy.

Leah S Larkey. 1998. Automatic Essay Grading Using Text Categorization Techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 90–95, Melbourne, Australia.

Lung-Hao Lee, Liang-Chih Yu, Kuei-Ching Lee, Yuen-Hsien Tseng, Li-Ping Chang, and Hsin-Hsi Chen. 2014. A sentence judgment system for grammatical error detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 67–70, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Teresa B. F. Martins, Claudete M. Ghiraldelo, Maria G. V. Nunes, and Osvaldo N. Oliveira Junior. 1996. *Readability formulas applied to textbooks in brazilian portuguese*. Instituto de Ciłncias Matemticas de So Carlos-USP, São Carlos, Brazil.

E. Martins. 2000. *Manual de redação e estilo*. O Estado de São Paulo.

Courtney Napoles and Chris Callison-Burch. 2015. Automatically scoring freshman writing: A preliminary investigation. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 254–263, Denver, CO, USA.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552, Beijing, China.

Alla Rozovskaya and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

Wei Song, Ruiji Fu, Lizhen Liu, and Ting Liu. 2015. Discourse Element Identification in Student Essays based on Global and Local Cohesion. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2255–2261, Lisbon, Portugal.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 46–56, Doha, Qatar.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-Independent Features for Automated Essay Grading. pages 224–232, Denver, CO, USA.

Table 6: Kappa results for Feature Analysis

Aspect	Feature Category	Feature Removed	Kappa
Final Grade	Most Relevant Features	Average word Length	0.3890
		Flesh Score	0.4010
		Vocabulary Level	0.4059
	Least Relevant Features	Discourse markers per #Sentence	0.4259
		Count of first Person	0.4262
		Count first Person per #Sentence	**0.4320**
		Full feature set	0.4245
Understanding the Task	Most Relevant Features	Flesh Score	0.1452
		#enclise / #sentences	0.1655
		#spelling errors	0.1655
	Least Relevant Features	#grammar errors	0.1868
		#style errors / #sentences	0.1878
		#first person use / # sentences	**0.1885**
		Full feature set	0.1817
Organization of Information	Most Relevant Features	Similarity with prompt	0.2496
		Average word length	0.2581
		#style errors / #sentences	0.2605
	Least Relevant Features	#long sentences	0.2788
		#demonstrative pronoun / # sentence	0.2799
		#first person use / #sentence	**0.2817**
		Full feature set	0.2728
Knowing Argumentation	Most Relevant Features	#spelling errors / #tokens	0.2438
		#style errors / #sentences	0.2441
		Flesh Score	0.2456
	Least Relevant Features	#enclise / #sentences	0.2773
		Average Word Length	0.2784
		#grammar errors	**0.2849**
		Full feature set	0.2668
Solution Proposal	Most Relevant Features	Average word length	0.1048
		Flesh score	0.1192
		#discourse markers	0.1240
	Least Relevant Features	#grammar errors / #Tokens	0.1586
		#tokens	0.1593
		#first person use	**0.1655**
		Full feature set	0.1655
Formal Language	Most Relevant Features	Flesh Score	0.3060
		#grammar errors / #tokens	0.3138
		#spelling mistakes	0.3248
	Least Relevant Features	#long sentences	0.3396
		#discourse markers	0.3396
		#demonstrative pronouns	**0.3429**
		Full feature set	0.3351

Literal or idiomatic? Identifying the reading of single occurrences of German multiword expressions using word embeddings

Rafael Ehren
Dept. of Computational Linguistics
Heinrich Heine University
Düsseldorf, Germany
Rafael.Ehren@hhu.de

Abstract

Non-compositional multiword expressions (MWEs) still pose serious issues for a variety of natural language processing tasks and their ubiquity makes it impossible to get around methods which automatically identify these kind of MWEs. The method presented in this paper was inspired by Sporleder and Li (2009) and is able to discriminate between the literal and non-literal use of an MWE in an unsupervised way. It is based on the assumption that words in a text form cohesive units. If the cohesion of these units is weakened by an expression, it is classified as literal, and otherwise as idiomatic. While Sporleder an Li used *Normalized Google Distance* to model semantic similarity, the present work examines the use of a variety of different word embeddings.

1 Introduction

Non-compositional multiword expressions (MWEs) still pose serious issues for a variety of natural language processing (NLP) tasks. For instance, if you use the free machine translation service Google Translate to translate example[1] (1-a) from English to German, according to the translation (1-b) the stabbing (luckily for John) doesn't cause his immediate death, but him literally kicking a bucket.

(1) a. Because John was stabbed, he kicked the bucket.
'Because John was stabbed, he died.'

b. Weil John erstochen wurde, trat er den Eimer.
'Because John was stabbed, he stroke a pail with his foot.'

Although not an absolutely impossible scenario, the context strongly suggests that *kicked the bucket* is not meant literally in (1-a) and therefore a literal translation is not the desired one.

Such errors illustrate the necessity for methods which automatically identify occurences of idiomatic MWEs when there is also a literal counterpart. Thus, there are actually two different identification tasks:

1. Determine wheter an MWE can have an idiomatic meaning;

2. Determine which of the two possible meanings, namely the literal and the idiomatic one, an MWE has given a specific context.

For example (1-a) this would mean to first figure out whether *kick the bucket* has another meaning than 'to strike a pail with one's foot' and then to decide which meaning it has in the context of the sentence. This paper is mainly concerned with the second task.

The method presented in this paper was inspired by the work of Sporleder and Li (2009) and is based on the assumption that words and sentences in a text are not completely independent of each other regarding their meaning, but form topical units. This relatedness between words is termed *lexical cohesion*. Sequences of words which exhibit a cohesive relationship are called *lexical chains* (Morris and Hirst, 1991). The intuition behind the approach is that idioms weaken this cohesion, because they often contain elements that are used in a figurative sense and thus do not "fit" into their contexts. If, for example, the MWE

[1] All of the examples presented in this paper were invented by the author.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 103–112, Valencia, Spain, April 3-7 2017.

break the ice is used in a literal sense, it will very likely co-occur with terms that are topically related like *snow, water, iceberg, etc.* This is usually not the case for the idiomatic use of *break the ice*. Consider the following example:

(2) For his future bride's sake he wanted to break the ice between him and his prospective parents-in-law before the wedding.

In (2), the expression *ice* appears with words (*wife*, *parents-in-law*, *wedding*) that do not belong to the same topical field as the literal meaning of *ice* and therefore it is not part of the dominating lexical chain.

Sporleder and Li made use of this fact and built cohesion-based classifiers to automatically distinguish between the literal and idiomatic version of an MWE. Following Sporleder and Li, we also implemented a classifier based on textual cohesion, albeit using a different measure for semantic similarity. While Sporleder and Li relied on *Normalized Google Distance* (NGD), a measure that uses the number of results for a search term as a basis, different word embeddings[2] were used in the context of this work. Word embeddings seemed like a more promising way of representing the meaning of words since a plain co-occurence-based approach like the NGD has some considerable limitations as we will discuss in section 3.2. Furthermore, a comparison of different types of embeddings was conducted where it became apparent that the implemented vector spaces are not all equally well suited for the task at hand. The task was conducted with a total of three different vector spaces and some achieved better results than others. Finally the best performing vector space was used to compare the effect of different window sizes around the MWE.

2 Related Work

Hirst and St-Onge (1998) followed the notion that words in a text are cohesively tied together and used it to detect and correct malapropisms. A malapropism is the erroneous use of a word instead of a similar sounding word, caused by a typing error or ignorance of the correct spelling. For instance: *It's not there fault*. In this sentence the adverb *there* is mistakenly used in place of the possessive determiner *their*. Since they are correctly spelled, malapropisms cannot be detected by spelling checkers that only check the orthography of a word. To tackle this problem, Hirst and St-Onge represented context as lexical chains and compared the words that did not fit into these chains with orthographically similar words. Semantic similarity was determined using WordNet.

Sporleder and Li (2009) were inspired by Hirst and St-Onge's method and applied it to MWEs, which they treated analogously to malapropisms. In their experiments the idiomatic version of an MWE is equivalent to a malapropism, because it usually does not participate in the lexical chains constituting the topic(s) of a text. Accordingly the literal sense of an MWE would be the correct word if we stay within the analogy. However, in contrast to Hirst and St-Onge, they did not rely on a thesaurus to model semantic similarity, but on NGD. As already stated in the introduction, NGD is a measure for semantic similarity that uses the number of pages returned by a search engine as a basis and is calculated as follows:

$$NGD(x,y) = \frac{max\{logf(x),logf(y)\} - logf(x,y)}{logN - min\{logf(x),logf(y)\}}$$

The number of pages for the search terms *x* and *y* are given by *f(x)* and *f(y)*, the number of pages containing *x AND y* by *f(x,y)*. *N* denotes the total number of web pages indexed by the search engine. If we take a look at the numerator we can see that it gets smaller the more often the two terms occur together. So an NGD of *0* means *x* and *y* are as similar as possible, while they get a score of greater or equal to *1* if they are very dissimilar.

With the NGD as a measure of semantic similarity, Sporleder and Li implementend two unsupervised cohesion-based classifiers that had the task to discriminate between the literal and non-literal use of an MWE. One of these classifiers did this based on the question whether a given MWE participated in one of the lexical chains in a text. If it did, the MWE was labeled as literal, if not, as idiomatic. The other classifier built *cohesion graphs* and made this decision based on whether the graph changed when the expression was part of the graph or left out (cohesion graphs will be elucidated in section 3.3).

Katz and Giesbrecht (2006) also examined a method to automatically decide whether a given MWE is used literally or idiomatically. Their method relied on word embeddings which were

[2]Word embedding is a collective term to denote the mapping of a word to a vector.

obtained through *Latent Semantic Analysis* (LSA). The experiment was conducted as follows: In a first step, Katz and Giesbrecht annoted for 67 instances of the German MWE *ins Wasser fallen* according to whether they were used literally or non-literally in their respective context.[3] Subsequently they generated a vector for the literal and a vector for the idiomatic use of the expression. In order to determine the meaning of the MWE with regard to the context, a nearest-neighbour classification was performed.

3 Setup

3.1 Lexical Cohesion

The term cohesion describes the property of a text that its items are not independent from one another, but somehow "tied together". Cohesion manifests itself in three different ways: back-reference, conjunction and semantic word relations (Morris and Hirst, 1991). Back-reference is usually realised through the use of pronouns (***Sarah*** *went to the dentist.* ***She*** *had a toothache.*). Conjunctions link clauses together and explicitly interrelate them (*John went home,* ***because*** *he was drunk*). But the only manifestation of cohesion significant for the present work are the semantic relations between the words in a text, i.e. the lexical cohesion. Lexical cohesion can be divided into five classes (Morris and Hirst, 1991; Stokes et al., 2004):

1. Repetition: *Kaori went into the* ***room****. The* ***room*** *was dark.*

2. Repetition through synonymy: *After a short rest Sally mounted her* ***steed****. But the* ***horse*** *was just too tired to go on.*

3. Repetition through specification/generalisation: *Shortly after he ate the* ***fruit****, his stomach began to cramp badly. It seemed that the* ***apple*** *was poisoned.*

4. Word association through a systematic semantic relationship (e.g. meronymy): *The* ***team*** *seemed unbeatable at that time. Already when the* ***players*** *went out on court, they put the fear of god in their opponents.*

5. Word association through a nonsystematic semantic relationship: *The* ***party*** *started at sunset. They* ***danced*** *till sunrise.*

[3]The literal meaning is 'to fall into the water', the idiomatic meaning is 'to fail to happen'.

Semantic relations like antonymy (*quiet*, *loud*), hyponymy (*bird*, *sparrow*) or meronymy (*car*, *tire*) are classified under systematic relationships. However, it is not always possible to specify the systematics behind a relationsip holding between two words (*party*, *to dance*). But for our purpose, it is not really necessary to identify the exact semantic relation, one only has to recognize that there is one. Even if we can't state what relation holds between *party* and *to dance*, we know that they are topically, and thus semantically, close.

Sequences of words exhibiting the forms of lexical cohesion listed above are referred to as lexical chains. These sequences, which can be more than two words long and cross sentence boundaries, span the topical units in a text (Morris and Hirst, 1991). In other words, they indicate what a text is about. That is why lexical chains can play an important role in text segmentation and summarization. The following example shows such a cohesive chain:

(3) When the ice finally broke the ice bear jumped off his floe into the ocean and fled. The icebreaker was designed to cut through the thickest ice, but soon it showed that even this huge ship could not withstand the unforgiving cold of the arctic. They had backed the wrong horse.

If we consider only the nouns in example (3) a possible lexical chain would be *ice*, *ice bear*, *floe*, *ocean*, *icebreaker*, *ice*, *ship*, *cold*, *arctic*. It indicates that the text segment is about the act of breaking sea ice. The lexical cohesion shows itself by repetition through generalisation (*icebreaker*, *ship*), repitition (*ice*, *ice*) and word association through unsystematic semantic relationships (e.g. *cold*, *arctic*). The only noun arguably not linked to any of the other words by a semantic relation and hence not participating in the cohesive chain is *horse*, the noun component of the idiomatic expresion *to back the wrong horse*. One could maybe argue that *horse* and *ice bear* share some semantic content since they are both four-legged mammals, but apart from that the case is pretty clear: *horse* is not part of the topical unit which is about the act of breaking sea ice. Therefore it's possible to conclude that *back the wrong horse* is not meant literally in this context.

Thus by looking for missing cohesive links one

is able to detect idiomatic readings of MWEs. In order to automatize this process, it is necessary to measure the semantic relatedness of two words. And to do that, it is in turn necessary to first model the meaning of words.

3.2 Word Embeddings

For their experiments Sporleder and Li (2009) modelled the semantic similarity of words in terms of the NGD. The advantage of the NGD is that no corpus can compare in size and up-to-dateness to the (indexed) web, which means that information regarding the words one is looking for is very likely to be found (Sporleder and Li, 2009).

Nevertheless, the method has some drawbacks. As Sporleder and Li state themselves, the returned page counts for the search terms can be somewhat unstable which is why they used Yahoo to obtain the web counts instead of Google because the former delivered more stable counts. Furthermore they had to leave out very high frequency terms because neither the Google nor the Yahoo API would deliver reliable results for those. But these are only minor issues compared to the fact that NGD is not the most sophisticated way of representing the semantics of words. The NGD reduces semantic similarity to the question of how often two terms occur together in a specific context relative to their total frequency. Although this simplification works surprisingly well, we will see herinafter that it has its limitations.

The basis for the representation of word meaning with distributional patterns is the distributional hypothesis. It states that words that occur in similar contexts have similar meanings. Or as John Rupert Firth prominently phrased it:

> "You shall know a word by the company it keeps!" (Firth, 1957, p. 11)

As an example, Firth gives the term *ass* which, according to him, is in familiar company with phrases like *you silly...*, *he is a silly...* or *don't be such an...* Not only would English speakers be able to guess with a certain probability which term they had to fill in for the dots, but other guesses presumably would fall on semantically similar words like *jerk*, *fool* or *idiot*. The validity of the distributional hypothesis and the fact that people only need a very small context window to infer the meaning of a word has been shown in different experiments (Rubenstein and Goodenough, 1965; Miller and Charles, 1991).

From the distributional hypothesis one can conclude that the semantic similarity of words does not manifest itself only through co-occurence (as the NGD simplifies), but also through shared neighbourhood. It might even be the case that some semantically very similar words appear less often together than one would expect, for example if a synonym is used to the exclusion of the other. Sahlgren (2006) did an experiment which strengthens this suspicion. He created two different representations of word meaning in form of vector spaces[4], one with a syntagmatic use of context and one with a paradigmatic use of context[5]. Then Sahlgren conducted the TOEFL synonym test[6] with both vector spaces and found that the paradigmatic word space achieved better results (75%) than the syntagmatic word space (67.5%). Sahlgren furthermore states that LSA performed on word-document matrices increases the results of TOEFL experiments because it reveals the "hidden" concepts behind words and thus relates words which do not co-occur, but appear in similar documents. This way, according to Sahlgren, a paradigmatic use of contexts is approximated. This shows that methods relying only on the co-occurence of words (syntagmatic relations) like the NGD are limited when it comes to the representation of word meaning. For that reason it seems more promising to model semantic relatedness with word embeddings, specifically word embeddings that represent syntagmatic **and** paradigmatic relations between words.

Word embeddings that incorporate a paradigmatic use of context by design are those who originate from the construction of a word-context matrix. But like documents in a term-document matrix, words in the word-context matrix are still only represented by bag-of-words. That is why structural vector space models (VSM) of word meaning were developed. These models, as one can already guess from the name, contain structural information about the words in the corpus,

[4]Words were represented by context vectors, NGD was not used in the experiment. But as it is the case with NGD one of the representations was created only considering co-occurence counts in a specific context region.

[5]A syntagmatic relation holds between to words that co-occur together, a paradigmatic relation holds between to words that share neighbours (i.e. they are potentially interchangeble).

[6]The TOEFL synonym test is a test were the testee has to choose the correct synonym for a given word out of four candidates (e.g. target word: levied; candidates: imposed, believed, requested, correlated; correct answer: imposed).

e.g. grammatical dependencies (Padó and Lapata, 2007). A model enriched with such information would, for example, be able to capture the fact that *the dog* is the subject and does the biting in the sentence *the dog bites the man*. A dimension of the word *dog* could thus be *sbj_intr_man*. The hope is that these models do a better job at representing semantics, because they take word order into account and ensure that there is an actual lexico-syntactic relation between the target and the context word and not only a co-occurence relationship.

An alternative to the "classic" count-based approach for the creation of word embeddings are skip-gram and continuous bag-of-words (CBOW). Skip-gram and CBOW, often grouped under the term *word2vec*, are two shallow neural networks which are able to create low-dimensional word embeddings from very large amounts of data in a relatively short amount of time. These two properties paired with the fact that the resulting word representations perform really well explain why word2vec has gained a lot of traction since Mikolov et al. (2013a; 2013b) presented it in 2013. In contrast to the "common" way of creating word embeddings by first constructing a word-context matrix of high dimensionality and then reducing the dimensions with LSA, word2vec creates low-dimensional vectors right from the start. This is possible, because skip-gram and CBOW do not count co-occurences in the corpus, but try to predict words. The skip-gram model tries to predict the neighbours of a word *w*, while CBOW tries to predict *w* from its neighbours. The intuition behind this approach is that a representation of a word that is good at predicting its surrounding words is also a good semantic represenation since words in similar contexts tend to have similar meanings (Baroni et al., 2014).

Levy et al. (2015) succeeded in showing that the perceived superiority of word2vec over traditional count-based methods (Baroni et al., 2014) is not founded in the algorithms themselves, but in the choice of certain parameters (Levy et al. call them "hyperparameters") which can be transferred to traditional models. Furthermore they showed that skip-gram with negative sampling (SGNS) implicitly generates a word-context matrix whose elements are Pointwise Mutual Information (PMI)[7] values shifted by a global constant. Hence, the data basis for word2vec and for the conventional methods is maybe not that different after all.

3.3 Experimental setup

To disambiguate between the literal and non-literal meaning of German MWEs it was of course necessary to first find instances of such MWEs. Those instances (along with the containing paragraphs) were automatically extracted from the TüPP-D/Z (Tübinger partiell geparstes Korpus - Deutsch/Zeitung)[8] corpus, a collection of articles from the German newspaper *die tageszeitung* (taz) from the years 1986 – 1999. Then the instances were annotated by hand depending on whether their readings were literal or idiomatic.

The MWEs listed in table 1 were chosen, because they are a part of figurative language and have a literal and idiomatic meaning. The latter is not self-evident, since some figurative MWEs do not have a literal meaning due to their syntactic idiosyncrasy, e.g. *kingdom come* and *to trip the light fantastic*.[9] And even the ones who do are mostly used in an idiomatic sense as one can see from the total count in table 1. 85% of the instances were used idiomatically.

MWE	Literal	Idiomatic	Total
jmdn. auf den Arm nehmen	19	31	50
das Eis brechen	3	82	85
etw. auf Eis legen	1	49	50
die Fäden ziehen	9	189	198
aufs falsche Pferd setzen	2	55	57
mit dem Feuer spielen	8	86	94
gegen den Strom schwimmen	1	60	61
die Kastanien aus dem Feuer holen	0	46	46
in den Keller gehen	28	63	91
im Regen stehen	20	80	100
den richtigen Ton treffen	30	80	110
in Stein gemeißelt sein	8	4	12
unter den Teppich kehren	0	75	75
ins Wasser fallen	46	124	170
das Wasser bis zum Hals stehen haben	17	75	92
total	192	1099	1291

Table 1: Instances of MWEs pulled form the corpus.

The annotation process revealed a considerable limitation of the cohesion-based method that was also mentioned by Sporleder and Li (2009): If the idiomatic reading is not isolated, but is lexically

[7] PMI is an association measure of two words. It is the ratio of the probability that the two words occur together to the probability that the two words appear independent of each other.

[8] *Tübingen Partially Parsed Corpus of Written German*

[9] Nunberg et al. point out that although "speakers may not always perceive the precise motive for the figure involved [...] they generally perceive **that** some form of figuration is involved" (1994, p. 492).

cohesive with regard to its context, the method obviously has to fail. But when does this happen? There were a few cases where an idiom did not stick out, because a whole metaphorical context was created around it. For example, one instance of the MWE *aufs falsche Pferd setzen* ('to back the wrong horse') was used together with other terms of the domain equitation to depict an unfortunate politician as a rider who falls from his horse. And sometimes it was the other way round. Some authors deliberately played with the ambiguity of an MWE by using it in a literal context with an idiomatic meaning (for example the fish who *swam against the tide*). Unfortunately this is a limitation one cannot overcome when using a cohesion-based method.

For the identification task a classifier was implemented that was based on the cohesion graphs of Sporleder and Li. An example for a cohesion graph is shown in Figure 1. In these undirected graphs nodes correspond to words and each node is connected with all other nodes. The edges are labeled with the cosine of the corresponding vectors. The cosine of an angle between two vectors is indicative for the semantic similarity of the words representend by those vectors. The larger the cosine (i.e. the smaller the angle), the more similar are these terms.

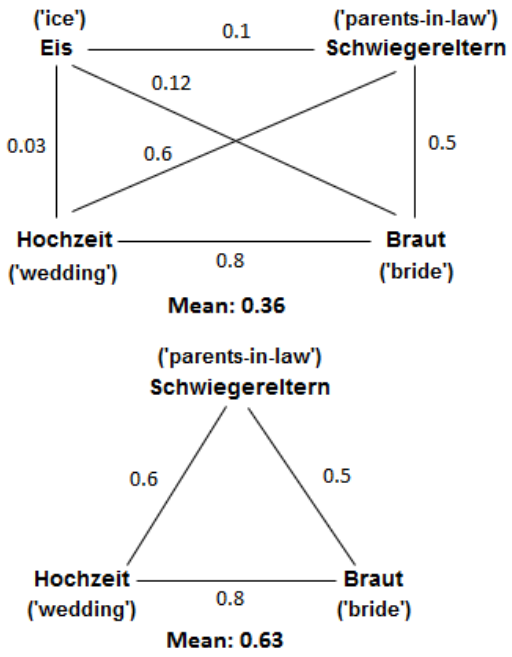


Figure 1: Example of two cohesion graphs with their respective mean cosine distance.

Figure 1 illustrates the identification process for example (2).[10] The graph at the top still contains the noun *Eis* component of the idiom *das Eis brechen*[11] and has connectivity mean of 0.36. In the graph at the bottom *Eis* was removed and the connectivity rose to a mean of 0.63. Since the cohesion between the words in the graph has increased, this is a sign for an idiomatic reading of the MWE.

The identification task was conducted as follows: First the paragraphs containing the instances of MWEs were reduced to only nouns (this will be explained later). Then the noun component of the MWE and a fixed number of neighbouring words were used to build a graph like in Figure 1. The similarity values were calculated by assigning the vector representations to the words from a vector lexicon and then calculating the cosine values of these vectors. After completing the graph the mean of the cosine values was calculated. After this the noun component of the MWE was removed from the graph and the mean was calculated again. If the mean got larger, the classifier labeled the instance of the MWE as *idiomatic*, if it stayed the same or got smaller, the instance was labeled as *literal*.

To test the impact of the different approaches on the representation of semantic similarity both types of VSMs, unstructured and structured, were employed in the experiments. Because the unstructured model did outperform the structured one, another unstructured model was built using different parameters to check, whether the performance could further be enhanced. Thus, a total of three different vector lexicons were used.

The first vector lexicon used was the German version of the Distributional Memory framework (DM) by Padó and Utt (2012). DM, originally designed for English by Baroni and Lenci (2010), is a structured distributional semantics model that includes grammatical dependencies. In contrast to the common approach to collect the data in a matrix, DM gathers it in a third-order tensor[12], i.e. in form of weighted word-link-word tuples (for ex-

[10] The nodes correspond to the nouns in (2). Since the experiments were conducted on the basis of a German corpus, the node labels are the respective German terms for *ice*, *parents-in-law*, *wedding* and *bride*.

[11] 'to break the ice'

[12] Tensors are generalisations of vectors and matrices. A first-order tensor is a vector, a second-order tensor a matrix and a third order tensor a three-dimensional array (Erk, 2012).

ample (*soldier*, *sbj_intr*, *talk* 5.42)). The tensor makes it possible to create different matrices on demand: word × link-word, word-word × link, word-link × word and link x word-word. For the purpose of this experiment a word × link-word matrix was generated since we want to compare the semantic similarity of single words. Then singular value decomposition (SVD)[13] was applied to the matrix to reduce the dimensions of the word vectors to 300.

The second vector lexicon was created with the word2vec tool on the basis of DECOW14, a German gigatoken web corpus provided by the COW (COrpora from the Web) initiative led by Felix Bildhauer and Roland Schäfer at Freie Universität Berlin (Schäfer and Bildhauer, 2012). The word embeddings generated by word2vec had a dimensionality of 100.

Last but not least, a third vector lexicon was created using the hyperwords tool provided by Omer Levy, also with the DECOW14 corpus as a basis. This tool incorporates the lessons learned of Levy et al. (2015) which were shortly presented in section 3.2. The word embeddings generated by hyperwords had 500 dimensions.

The decision to only include nouns in the identification process was made to significantly reduce the size of the vector lexicons and thereby the computational costs. Nouns were chosen, because they are considered to be the best topic indicators in a text.

All three vector lexicons were tested with a window of size six around the MWE.[14] Subsequently the best performing vector lexicon was tested with context windows of size two and size ten to examine the effect of the window size on the performance.

4 Results

The baseline for the experiments was a classifier that labeled all instances with the majority class. Thus, the accuracy, for example, would be 85.13% because 85.13% of the instances are idiomatic. Since we made the assumption that word embeddings are better suited for the presented method than the NGD, the NGD would of course have been a more natural baseline. Unfortunately, getting the required data proved to be not that easy because the access to the search APIs of the major search engines seems to be more restricted than a few years ago.

Table 4 shows the results for the three vector lexicons with a context window of 6. With an accuracy of 63.35% DM showed by far the worst performance, falling short of the baseline by a large margin. The reason might be that while the NGD only considers syntagmatic relations between words (i.e. the question if they co-occur), DM seems to have its focus on paradigmatic relations. This would explain why words like *France - Italy* (0.84)[15], *president - Pope* (0.78) and *minister of defence - general* (0.77) are pretty close in this word space, whereas terms like *murder - court* (0.058), *president - USA* (0.078) and *city - border* (0.047) are very far apart, though clearly topically related. Words that build a paradigm exhibit a substitutional relationship which means that one word can potentially replace the other in a specific context (e.g. *The president/Pope gave a speech.*). And if a word can be replaced by another this in turn means that they have to be attributionally similar which appears to be exactly the kind of similarity DM represents. This is bad news for the task at hand, since lexical cohesion, as we saw, not only incorporates attributional similarity, but all kinds of relations. However, words that are connected by a nonsystematic relationship are very dissimilar to each other according to DM. This could indicate that structural distributional semantics models (at least the ones that rely on grammar dependencies) are not the best solution for cohesion-based tasks.

Word2vec on the other hand delivered with an accuracy of 81.03% the best performance for a context window size of 6 (but still falling below the baseline by ca. 4%). This is in accordance with the above presented suspicion that a structured model is not a good fit for the conducted experiments. After all word2vec respectively skip-gram (which was used for the experiment) is an unstructered model. In contrast to DM, word2vec not only seems to model attributional similarity, e. g. *Apfel - Birne*[16] (0.8), but also topical relat-

[13] SVD is a dimensionality reduction technique. Through SVD a matrix is decomposed in three matrices whose dimensions are reduced to a desired number. The matrices originating from this process approximate the original matrix. This is possible because the remaining dimensions are the principal components of the data, i.e. they convey the most information.

[14] The number of neighbouring words that were included in the cohesion graphs along with the noun component of the idiom.

[15] In the parentheses behind the word pairs are the cosine values.

[16] *Apfel* means 'apple', *Birne* means 'pear'.

	DM			Word2vec			Hyperwords		
MWE	**Pre**	**Rec**	**Acc**	**Pre**	**Rec**	**Acc**	**Pre**	**Rec**	**Acc**
jmdn. auf den Arm nehmen	53.33	26.67	39.58	81,25	86,67	79,20	61,34	90,00	58,33
das Eis brechen	95.56	55.13	54.32	98,53	85,60	85,20	97,33	93,59	91,36
etw. auf Eis legen	100	50.00	51.06	97,87	100	98,00	97,87	100	97,87
die Fäden ziehen	96.93	86.81	84.82	97,42	82,51	81,25	96,49	90,66	87,96
aufs falsche Pferd setzen	93.94	62.00	59.62	98,08	100	98,10	96,23	100	96,23
mit dem Feuer spielen	91.25	91.25	84.09	95,89	87,50	85,23	97,06	82,50	81,82
gegen den Strom schwimmen	98.15	92.98	91.38	100	87,93	88,14	100	69,64	70,18
die Kastanien aus dem Feuer holen	100	100	100	100	68,89	68,89	100	68,89	68,89
in den Keller gehen	71.43	8.06	32.18	85,94	88,71	81,61	78,67	95,16	78,16
im Regen stehen	80.60	72.00	64.21	87,80	93,51	84,54	87,01	88,16	80,21
den richtigen Ton treffen	77.78	18.42	37.14	82,43	78,21	71,96	82,09	71,43	67,92
in Stein gemeißelt sein	50.00	25.00	63.64	42,86	75,00	58,33	37,5	75,00	50,00
unter den Teppich kehren	100	45.59	45.59	100	92,86	92,86	100	91,30	91,30
ins Wasser fallen	74.10	86.55	67.90	85,22	81,67	76,22	81,89	87,39	76,69
das Wasser bis zum Hals stehen haben	81.82	88.73	74.71	92,00	63,89	65,91	86,30	87,50	78,41
total	84.33	60.61	63.35	89,69	84,86	81,03	86.65	86,08	78,36

Table 2: Results for the three different vector spaces with a context window of size 6.

edness as is shown in Figure 1. A *wedding* and a *bride* do not have much in common in terms of their properties (one is an event, the other is a human being), but they are undoubtedly topically close as word2vec correctly assumes (0.8).

The performance of hyperwords (78.63% accuracy) is comparable to that of word2vec which is not very surprising since it also uses SGNS only with different parameter settings.[17]

The best model, word2vec, was then used to examine the effect of different context window sizes on the performance. At first, a very narrow window of size 2 was tested to check whether the two closest neighbours[18] are sufficient to identify the idiomatic reading of an MWE. The results seen in table 3 suggest they are not. With 63.26% accuracy it performs as badly as the DM model with a context window of 6.

Subsequently a broader window of size 10 was used while conducting the task. In contrast to the narrow window it performed well and achieved with an accuracy of 85.67% (see table 4) the highest score of the experiment, surpassing the accuracy baseline by a slight bit. But since we want our classifier to perform well on both classes, idiomatic and literal, it is important to also have a look at the precision (90.47%) which surpasses the baseline by more than 5%. The good performance

[17]Hyperwords offers two different possibilites: the 'old way' of creating a word-context matrix reduced with SVD, and SGNS. We used SGNS in the experiments.

[18]Reminder: The noun component of the MWE is in the focus of the window.

MWE	**Pre**	**Rec**	**Acc**
jmdn. auf den Arm nehmen	76,00	61,29	64,00
das Eis brechen	98,25	70,00	69,88
etw. auf Eis legen	97,78	89,80	88,00
die Fäden ziehen	96,50	58,20	58,08
aufs falsche Pferd setzen	95,56	78,18	75,44
mit dem Feuer spielen	98,11	61,90	64,13
gegen den Strom schwimmen	100	63,33	63,93
die Kastanien aus dem Feuer holen	100	8,89	8,89
in den Keller gehen	85,71	76,19	74,73
im Regen stehen	88,57	77,50	74,00
den richtigen Ton treffen	85,48	66,25	67,27
in Stein gemeißelt sein	60,00	75,00	75,00
unter den Teppich kehren	100	72,97	72,97
ins Wasser fallen	84,54	66,13	66,47
das Wasser bis zum Hals stehen haben	81,82	12,00	26,09
total	89,89	62,51	63,26

Table 3: Results for the word2vec vector space with a context window of size 2.

compared to the other results indicates a correlation between the size of the context window and the performance of the model.

5 Conclusion

The experiments conducted in the course of this work show that the presented method generally produces good results if a suitable vector lexicon is used and the context window is large enough. These results could probably further be improved if different parameters are optimized. It is possible that the model would achieve even better results by including verbs in the cohesion graphs in addition to nouns since they are also good topic indicators. In addition, it would be interesting to see

MWE	Pre	Rec	Acc
jmdn. auf den Arm nehmen	86,67	92,86	86,36
das Eis brechen	100	88,89	89,33
etw. auf Eis legen	97,73	100	97,73
die Fäden ziehen	97,30	86,75	85,14
aufs falsche Pferd setzen	97,83	100	97,83
mit dem Feuer spielen	95,65	90,41	87,65
gegen den Strom schwimmen	97,67	91,30	89,36
die Kastanien aus dem Feuer holen	100	85,37	85,37
in den Keller gehen	87,30	94,83	86,42
im Regen stehen	86,84	97,06	85,71
den richtigen Ton treffen	85,71	89,55	81,52
in Stein gemeißelt sein	50,00	75,00	60,00
unter den Teppich kehren	100	96,77	96,77
ins Wasser fallen	82,57	83,33	74,66
das Wasser bis zum Hals stehen haben	91,80	84,85	81,25
total	90,47	90,46	85,67

Table 4: Results for the word2vec vector space with a context window of size 10.

up to which point an enlargement of the context window results in a better performance.

For further future work, it would be desirable to test if the method could be used to automatically discover non-compositional MWEs when combined with a statistical approach. First, with help of a measure of association one could generate a candidate list of statistically idiomatic MWEs whose instances are then examined for lexical cohesion with respect to their contexts. This way, it may be possible to discriminate between institutionalized phrases and non-compositional MWEs.

6 Acknowledgements

I am grateful to my supervisors Laura Kallmeyer and Timm Lichte for their valuable feedback and guidance throughout this work. In addition I would like to thank Sebastian Padó, Jason Utt, Felix Bildhauer and Roland Schäfer for the provided ressources and the three anonymous reviewers for their comments on this paper.

References

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

John Rupert Firth. 1957. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*, pages 1–32. Blackwell, Oxford.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.

Graham Katz and Eugenie Giesbrecht. 2006. Automatic identification of non-compositional multiword expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19, Sydney, Australia, July. Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, 17(1):21–48.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language*, pages 491–538.

Sebastian Padó and Jason Utt. 2012. A distributional memory for german. In Jeremy Jancsary, editor, *KONVENS*, volume 5, pages 462–470. ÖGAI, Wien, Österreich.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Magnus Sahlgren. 2006. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. Ph.D. thesis, Stockholm University, Stockholm, Sweden.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul. ELRA.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece, March. Association for Computational Linguistics.

Nicola Stokes, Joe Carthy, and Alan F. Smeaton. 2004. Select: a lexical cohesion based news story segmentation system. *AI Communications*, 17(1):3–12.

Evaluating the Reliability and Interaction of Recursively Used Feature Classes for Terminology Extraction

Anna Hätty
Robert Bosch GmbH
Anna.Haetty@
de.bosch.com

Michael Dorna
Robert Bosch GmbH
Michael.Dorna@
de.bosch.com

Sabine Schulte im Walde
IMS, University of Stuttgart
schulte@
ims.uni-stuttgart.de

Abstract

Feature design and selection is a crucial aspect when treating terminology extraction as a machine learning classification problem. We designed feature classes which characterize different properties of terms, and propose a new feature class for components of term candidates. By using random forests, we infer optimal features which are later used to build decision tree classifiers. We evaluate our method using the ACL RD-TEC dataset. We demonstrate the importance of the novel feature class for downgrading termhood which exploits properties of term components. Furthermore, our classification suggests that the identification of reliable term candidates should be performed successively, rather than just once.

1 Introduction

Terms are linguistic units which characterize a specific topic domain. For example, in the area of Computational Linguistics *Parsing*, *Machine Translation* and *Natural Language Generation* are candidates for single and multi-word terms. Automatic Term Recognition (ATR) is the task of identifying such terms in domain-specific corpora. ATR is an Information Extraction subtask and is used i.a. for compiling dictionaries and for ontology population (Maynard et al., 2008). A typical ATR system comprises two steps: First, term candidates are selected from text, e.g. by extracting sequences which match certain part-of-speech (POS) patterns in text (c.f. Justeson and Katz, 1995). Secondly, term candidates are scored and ranked with regard to their unithood and termhood.

Unithood denotes to what degree a linguistic unit is a collocation. *Termhood* expresses to which extent an expression is a term, i.e to which extent it is related to domain-specific concepts (Kagueura and Umino, 1996). Among a large number of measures, association measures like *Pointwise Mutual Information* (PMI) (Church and Hanks, 1989) are used to determine unithood whereas term-document measures like *tf-idf* (Salton and McGill, 1986) are used to determine termhood. Such measures use distinctive characteristics of terms on how they and their components are distributed within a domain or across domains.

We address term extraction as a machine learning classification problem (c.f. da Silva Conrado et al., 2013). Most importantly, we focus on the interpretability of a trained classifier to understand the contributions of feature classes to the decision process. For this task, we use random forests to automatically detect the best features. These features are used to build simple decision tree classifiers.

For the classification, we use features based on numeric measures which are computed from occurrences of term candidates, its components and derived symbolic information like POS tags. We call these *distributional features*. The advantage of relying on such features is that they are simple to compute and easy to compare. By combining machine learning with those features we get a flexible system which only needs little further information to be applicable on different kinds of text. In this work, we investigate the contributions of the different features to term extraction and experimentally test with our system if these features are mutually supportive. We also point out the limit of a system solely relying on distributional features.

The paper is organized as follows. Section 2 introduces related work. The data used for training and evaluation is presented in Section 3, followed by the feature selection and classification method.

Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 113–121, Valencia, Spain, April 3-7 2017.

Our feature classes are motivated and defined in Section 4. In Section 5, we investigate the design of our models with a subsequent presentation of experiments and evaluation results in Section 6. In Section 7, we present a second experiment with term candidates which share a component to further explore their contribution to termhood.

2 Related Work

There are several studies investigating linguistic and numeric features, machine learning or a combination of both to extract collocations or terms. Pecina and Schlesinger (2006) combined 82 association measures to extract Czech bigrams and tested various classifiers. The combination of measures was highly superior to using the best single measure. Ramisch et al. (2010) introduced the *mwetoolkit* which identifies multi-word expressions from different domains. The tool provides a candidate extraction step in advance, descriptive features (e.g. capitalisation, prefixes) and association measures can be used to train a classifier. The latter ones are extended for multi-word expressions of indefinite length and only comprise measures which do not depend on a contingency table. Karan et al. (2012) extract bigram and trigram collocations for Croatian by relying on association measures, frequency counts, POS-tags and semantic similarities of all word pairs in an *n*-gram. They found that POS-tags, the semantic features and PMI work best. With regard to terms, Zhang et al. (2008) compare different measures (e.g. tf-idf) for both single- and multi-word term extraction and use a voting algorithm to predict the rank of a term. They emphasize the importance of considering unigram terms and the choice of the corpus. Foo and Merkel (2010) use RIPPER (Cohen, 1995), a rule induction learning system to extract unigram and bigram terms, by using both linguistic and numeric features. They show that the design of the ratio of positive and negative examples while training governs the output rules. Da Silva Conrado et al. (2013) investigate features for the classification of Brazilian Portuguese unigram terms. They use linguistic, statistical and hybrid features, where the context and the potential of a candidate representing a term is investigated. Regarding the features, they find tf-idf essential for all machine learning methods tested.

3 Data and Classification Method

3.1 Corpus and Gold Standard

The underlying data set for the experiments is the ACL RD-TEC 1.0[1], a corpus designed for the evaluation of terminology extraction in the area of Computational Linguistics (Zadeh and Handschuh, 2014). It extends ACL ARC, an automatically segmented and POS-tagged corpus of 10,922 ACL publications from 1965 to 2006. ACL RD-TEC adds a manual annotation of 22,044 valid terms and 61,758 non-terms. The term annotations are further refined with a labeling of terminology terms which are defined as means to accomplish a practical task, like methods, systems and algorithms used in Computational Linguistics. We take the valid terms as our gold standard terms. We cleaned the corpus by applying a language detection tool (*langdetect*[2]) to each sentence, in order to remove sentences which are too noisy. A drawback of the corpus is that about 42,000 sentences could not be connected to a document. Thus, if no document was found for a certain term, its term-document measures were set to a default value outside of a feature's range, or to an extreme value.

3.2 Feature Reduction and Classification

Unigrams, bigrams and trigrams which appear at least ten times in the text are extracted from the corpus as term candidates. For all candidates, features are computed (see Section 4). As a preprocessing step, a **random forest classifier** (Breiman, 2001) with 100 estimators is used for feature reduction. To prevent overfitting, each of these decision trees is trained on a subset of the data, and a randomly chosen subset of features (here the square root of the number of features) is considered for splitting a node. Considering all internal decision trees, the contribution of the features to the classification is evaluated and averaged. In this way, we get good estimates of the importances of each feature and can use them for feature reduction: the classifier returns the importance scores for the features, and feature selection is performed by only taking those features whose score is greater than the mean. Subsequently, a **decision tree classifier** (Breiman et al., 1984) is trained with those features that provide a single representation for the decisions. The training set

[1] http://atmykitchen.info/nlp-resource-tools/the-acl-rd-tec
[2] https://pypi.python.org/pypi/langdetect?

was balanced for terms and non-terms to prevent a bias in the classifier. In the first step, everything which is not marked as term is treated as non-term. We only allowed POS patterns also occurring in the term class and chose randomly to get a representative sample of non-terms. In the second step, we use the explicitly annotated non-term class.
Both classifiers produce binary decision trees and an optimized version of the CART algorithm[3] is used.
As split-criterium for the decision trees we used *entropy* and we only allowed trees to evolve up to five levels, since otherwise they overfit. In addition, trees are very difficult to understand when getting deeper than five levels and we explicitly chose decision trees because of their clear interpretability. For the interpretation and evaluation in the following, the construction of the final decision trees for each *n*-gram and their classification performances will be used.

4 Feature Classes

A salient attribute of terms is how they distribute in text. Our feature classes are motivated by three perspectives on that: a) measuring unithood involving the distribution of term candidates and their components, b) measuring termhood involving candidate term distributions in different texts and c) recursively measuring unithood and termhood of term candidate components independently of each other. Concerning the classes defined in the following, point a) is covered by the *association measures*, b) by *term-document* and *domain specificity measures* and c) by the *features of components*. In addition, we designed *count-based measures* and a *linguistic feature* to address unithood and termhood. However, we expect them to be weaker than the feature classes of a) and b) since they do not relate two distributions. They merely serve for filtering, ruling out very unlikely term candidates.

Term-Document Measures (TD) The term-document measures deal with the distribution of term candidates in certain documents and contrast it to their distribution in the whole corpus. It is assumed that terms appear more frequently in only a few documents. We include a range of features dealing with that contrast: variants of tf-idf (Salton and McGill, 1986), i.e. *tf-idf* (without logarithm), *tf-logged-idf* for the document in which the term candidate occurs most often. Furthermore, *corpus maximum frequency* and *corpus maximum frequency & term average frequency (cmf-taf)* as defined in Tilley (2008), and *term variance* and *term variance quality* as described in Liu et al. (2005) are used. Da Silva Conrado et al. (2013) describe the latter features as useful for term extraction. In addition, we experimented with features describing the relative occurrence of a term in a document or the corpus. For example, the percentiles of document or corpus frequencies are used as features, to which the frequency of the term under consideration can be assigned. Another example is the percentile of the document with the term candidate's first position. In the later experiments, these features are assigned little weight by the classifiers which is why we will not go into further detail regarding them.

Domain Specificity Measures (DS) Measures of domain specificity treat the occurrence of a term in a general corpus and relate it to its occurrence in a domain-specific one. As domain-specific corpus, we simply chose the document with the most frequent occurrence of a term candidate. By doing that, the problem is omitted that the vocabulary of these corpora differs too drastically due to aspects of style. As features *weirdness ratio for domain specificity, corpora-comparing log-likelihood (corpComLL), term frequency inverse term frequency (TFITF)* and *contrastive selection of multi-word terms (CSmw)* are used (as defined in Schäfer et al., 2015).

Association Measures (AM) Association measures express how strongly words are associated in a complex expression, they measure unithood. 27 association measures defined in Evert (2005) were computed for bigrams, for example *Local Mutual Information (LocalMI)* and *Maximum Likelihood Estimation (MLE)*. For trigrams, we selected nine association measures (*MLE, PMI, Dice, T-score, Poisson-Stirling, Jaccard,* χ^2*, Simple Log Likelihood* and *true MI*) which are described as useful for trigram association in Lyse and Andersen (2012), Ramisch et al. (2010) and the *nltk*-documentation [4].

Count-based Measures (Count) Wermter and Hahn (2006) compare co-occurrence frequencies

[3] http://scikit-learn.org/stable/modules/tree.html#tree

[4] www.nltk.org/_modules/nltk/metrics/association.html

and association measures and show that not association measures but only linguistically motivated features outperform frequency counts for collocation and terminology extraction. Therefore *frequencies* of the term candidates are included in the feature set. As described, we do not consider them as being as powerful as association measures (and they only play a minor role in our later models). The second count-based measure is *word length*.

Linguistic Feature (Ling) As linguistic feature, *Part-Of-Speech*-tags (POS) of the candidates are used to represent distributions over POS patterns.

Features of Components (Comp) The components of a term phrase have frequently played a role in termhood extraction (e.g. Nakagawa and Mori, 2003; Zhang et al., 2012). Our approach differs from the previous ones by adding all feature information of the candidate term components to the candidates's feature set. I.e., for bigrams the features of its unigrams, and for trigrams the features of its uni- and bigrams are included. The features will be characterized with the following scheme: [POSITION IN TERM]-[COMPONENT IS A UNI- OR BIGRAM]-[FEATURE]. Examples would be *0-uni-CSmw* denoting the CSmw-feature for the first word X in bigram XY or *1-bi-CSmw* denoting the CSmw-feature for second bigram YZ in trigram XYZ. *1-bi-POS != NN NN* expresses that the second bigram YZ in trigram XYZ does not consist of nouns.

Class	1	2,3	Feature Examples
TD	+	+	tf-idf, cmf-taf, term variance
DS	+	+	weirdness ratio, corpComLL, TFITF
AM	-	+	PMI, LocalMI, Chi2
Count	+	+	frequency, word length
Ling	+	+	POS pattern
Comp	-	+	0-uni-POS, 1-bi-tf-idf

Table 1: Overview of Feature Classes

An overview of the classes is given in Table 1. The labels 1, 2 and 3 in the table denote uni- to trigrams, + and - express if a class can be applied or not. For unigram terms (SWT) not all feature classes can be applied.

5 Inspecting the Models

Combining all previously mentioned features with our classification method (i.e. unigrams, bigrams and trigrams) provides three decision trees. For ease of visualization and interpretation, only the first three decision levels are shown in the following figures (Figures 1 to 3). The tree is only allowed to evolve further if the distinction between terms and non-terms could not be made to that point. Furthermore, splitting a node is stopped if there are less than 10 elements in a leaf for one of the classes (even if the tree limit has not been reached yet).

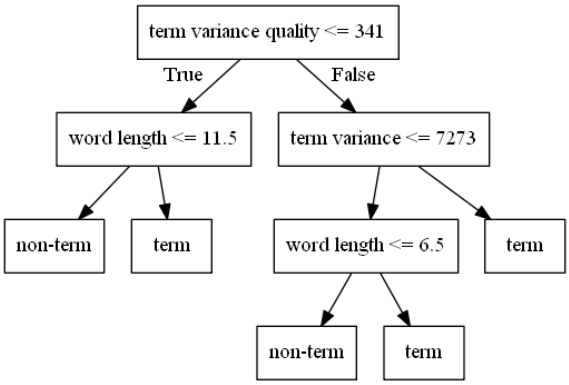


Figure 1: Decision Tree for Unigrams

Unigrams The decision tree for unigram classification based on 1608 unigram terms and non-terms is shown in Figure 1. Term variance quality and term variance best classify terms; In the resulting leaf node (rightmost node) 90% of the 324 elements are correct terms. When looking at the false positives in that node, it is striking that the few non-terms remaining in that class are unexpectedly "usual" (*'czech', 'newspaper', 'chain', 'travel', 'situation'*). The reason for this unexpected classification might result from the context in which the study is conducted: there might be papers which are limited to Czech data or only to newspaper texts.

The construction of the whole decision tree reveals that the classifier tries to identify clear-cut sets of terms using decision thresholds with extreme values. Following the path on the right-hand side, the subset of elements with the highest termhood scores is isolated. If the term-document measure values are not distinctive anymore (taking left branches) non-terms are singled out by filtering via word length. The less distinctive termhood measures are, the less word length is limited on filtering extremely short and therefore extremely unlikely term-candidates. This is an on-demand filtering step: term candidates are not only filtered in advance, but the threshold is adjusted to how

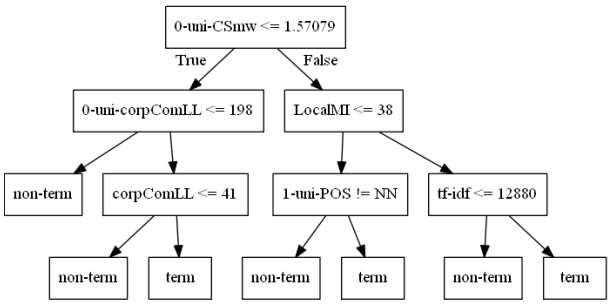


Figure 2: Decision Tree for Bigrams

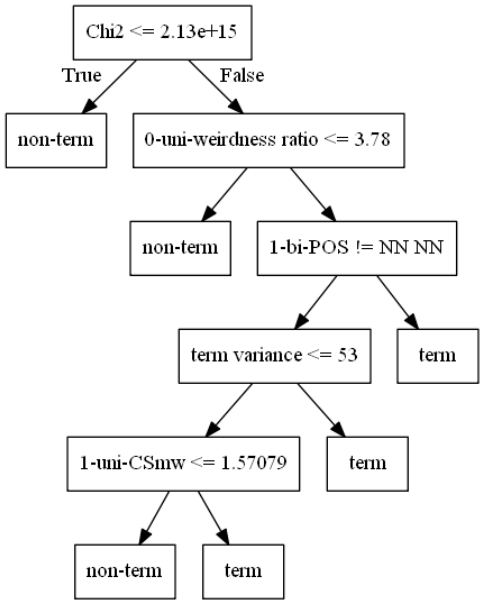


Figure 3: Decision Tree for Trigrams

significant the termhood measures are.

Bigrams The decision tree for the 10,562 extracted bigram candidates is depicted in Figure 2. Features for the first component like 0-uni-CSmw are good indicators for termhood. When inspecting how the bigrams are distinguished by the root node it seems that if the first word of a bigram is a general-language word, the whole bigram is unlikely to be a term. There are quite obvious examples like *this specification, the parser, a hurry* or *another expression* but also more interesting ones like *earlier paper, particular cluster* or *general scheme*. Nevertheless, in other term leaves there are still quite a few expressions whose first words are not terminological (e.g. *simple formalism, common description, good hypothesis*), so there is still room for improvement.

Trigrams The decision tree for trigram classification of 1706 trigram candidates is shown in Figure 3. The association measure χ^2 (*Pearson's chi-squared test*; c.f. Evert, 2005) is by far the most important feature here and the sets are nearly completely distinguished by that feature. Thus, unithood nearly merges to termhood here. Besides that, it is again striking that expressions with non-terminological first components are ruled out correctly by the system, e.g., *possible syntactic category*, *other natural language*, *new grammar formalism*. There are also misclassifications (false negatives) like *first order logic*. The rightmost path produces the purest right-most node compared to all previous ones for uni- and bigrams: 94% of the 636 elements are correct terms.

Comparison Across the decision trees different features dominate the tree, which shows that uni-, bi- and trigram terms behave differently and should be treated differently. Nevertheless, they have in common that the trees are dominated by termhood and unithood features and that features for filtering noise like POS patterns and word length occur lower in the tree. This supports the already mentioned claim that several filtering steps should be performed at different stages of the classification. As a second commonality, the trees combine features from various classes in their first decision steps. Especially in the rightmost path, in which terms are separated best in the experiments, term-document measures, association measures and domain-specificity measures of components are combined. This shows that features from different feature classes interact for achieving a good result.

6 Experiments and Results

Our system is implemented in Python. For the classifications, we used the *RandomForestClassifier* and the *DecisionTreeClassifier* which are included in the Python module *sklearn* (Pedregosa et al., 2011).

Baselines For each *n*-gram class, the best-working feature is chosen as a baseline. These are the root nodes of the decision trees for all features because these ones are chosen first, given that they make the best decision. The baselines are *term variance quality* for unigrams, *0-uni-CSmw* for bigrams and *Chi2* for trigrams.

Performance of Individual Feature Classes As a first evaluation step, the different feature classes are compared. For that, decision trees

are separately trained for each feature class. We do 10-fold cross-validation with a balanced set of terms and non-terms in every step. The performances of the different classes for unigrams, bigrams and trigrams are shown in Table 2. When considering the overall results (F1-score), it is striking that for bigrams and trigrams the component features (Comp) achieve the best score, middle-ranking groups are the count-based features (Count) and the linguistic feature (Ling), and the term-document (TD) and domain-specific features (DS) are in the lower area. This is quite a surprising result since these are the termhood features and therefore the ones to be expected to perform best. For unigrams, in contrast, term-document features and domain specificity are good indicators for classification. However, when considering precision, the domain specificity features lag behind. They do not seem to be competitive to term-document metrics in that respect. All in all, domain specificity features do not reach the expected performance here. This is an interesting result because when the domain specificity features are used for the components of an *n*-gram they appear in the upper part of the tree. We conclude that the features for domain specificity applied to components receive the unexpected application of downgrading the termhood of a term candidate if a component under consideration is unlikely to be terminological.

Feat. Class	TD	DS	Assoc	Count	Ling	Comp
Unigrams						
Precision	**0.75**	0.67	-	0.73	0.63	-
Recall	0.71	0.73	-	0.66	0.81	-
F1-Score	**0.72**	0.70	-	0.69	0.70	-
Bigrams						
Precision	0.72	0.65	0.72	**0.73**	0.67	**0.73**
Recall	0.71	0.79	0.65	0.79	0.88	0.88
F1-Score	0.71	0.71	0.68	0.76	0.76	**0.80**
Trigrams						
Precision	0.67	0.59	0.85	0.75	0.80	**0.88**
Recall	0.72	0.72	0.96	0.82	1.0	0.97
F1-Score	0.69	0.65	0.90	0.78	0.89	**0.92**

Table 2: Precision, Recall and F1-Scores for Feature Classes

Evaluating All Features As a last step, we evaluate if the combination of different features outperforms the best single feature. For that we do 10-fold cross-validation with a balanced set of terms and non-terms in every step. The results are shown in Table 3. All systems which combine features outperform the baselines. In addition, they also outperform the best systems which only use one feature class at a time (Table 2). All these improvements are significant, [5] except for the comparison of the overall model for trigrams to the model of its best-working class (*features of components*). This shows that a combination is not only superior to a baseline but also information from several classes is needed. Term recognition works best for trigrams and is most difficult for unigrams.

Method	Precision	Recall	F-score
Baseline	0.62	0.85	0.70
Unigrams	0.75	0.79	**0.77**
Baseline	0.60	0.89	0.72
Bigrams	0.78	0.87	**0.81**
Baseline	0.84	0.97	0.90
Trigrams	0.89	0.96	**0.93**

Table 3: Results

7 The Relevance of the Component Class

In the previous experiments we investigated how terms can be distinguished from candidates in the scientific text which are restricted by POS but which are otherwise randomly chosen. For bigrams and trigrams, the component class performs best. Since the components of candidate terms seem to have a major influence on their termhood, we further investigate the components. For that, candidates are not chosen randomly anymore, but are taken from the class explicitly annotated as non-terms by Zadeh and Handschuh (2014). The reason for this is that the elements of the provided annotated term and non-term expressions have identical components in many cases. Like that term candidates with components which are not uniquely terminological or non-terminological are used for training the classifier. Subsets of the classes are compared three times: Only those elements are allowed where either the first, the second or the third component (in case of trigrams) appears in both classes. The results are presented in Table 4.

The results indicate that a clearly terminological or non-terminological first component has more effect on the termhood of the whole expression than for the last component. If the first component is fixed and thus is not relevant for scoring termhood, results decrease.

[5] χ^2, $p<0.01$

	Bigrams			Trigrams		
Feature Class	**P**	**R**	**F1**	**P**	**R**	**F1**
last component	0.69	0.83	0.76	0.76	0.77	0.76
mid component	-	-	-	0.73	0.75	0.74
first component	0.66	0.70	0.68	0.73	0.71	0.72

Table 4: Results for identical elements for different components in term- and non-term class

This is also reflected in the decision trees: For identical heads, the most important feature is the component feature of the first unigram and of the first bigram. For identical modifiers no component feature is chosen as most important feature.

8 Discussion and Future Work

There are two main points why a system like ours only based on distributions reaches its limit. One aspect is the unexpected fluctuations of general-language terms shown especially for unigram term extraction. We found words being classified as terms because they often appear in the context of a special experimental setting. Secondly, our results show that it is harder for such a system to distinguish term candidates with shared components than to distinguish terms from a representative part of the other in-domain text as done in the first experiment (Table 3 vs. Table 4).
However, the advantages of our model suggest that it can be applied to extract terms from forum text, a topic which has not received much attention yet. The information used in the model, the features and their application on components of the term candidates, can be easily computed on the text and additional resources are not necessarily needed. Another advantage of our model is that it is dynamic. Uni-, bi- and trigrams are quite different in nature which is reflected in the models. It filters improbable term candidates by making several decision steps adapted to the data seen in training. Thus, we might not need a preprocessing step to filter good candidates. In both experiments, with and without an explicitly annotated non-term class, applying the features to components of the candidates improves the extraction. We find that especially the features for the first parts, mostly the modifier, are good dividers for the term and the non-term class. Since the number of non-terminologic modifiers (like judging adjectives) will be higher in forum texts, this aspect will be a further advantage.

9 Conclusion

In this work, term extraction was approached as a classification problem using uni-, bi- and trigram term candidates. We used a decision tree classifier to model term recognition with focus on the distribution of terms and of its components in text. Different classifier setups were compared: classifiers for the single best feature, different feature classes and a combination of all features. In each of those steps classification improves. Neither a feature class nor a special feature constantly dominates the classification in all models. The construction of the decision trees reveals that there is an interaction of features of different classes. Features from the most adequate classes to recognize terms, i.e. features which measure termhood and unithood, interact to find the purest term class.
The resulting decision trees from the experiments indicate that there should not be a rigid pipeline of two steps, where candidate extraction and filtering noise comes first, and subsequently the terms should be scored and ranked. Our results indicate that there should rather be an on-demand filtering step, where filtering is performed successively during the classification and the threshold for ruling out extremely unlikely candidates is adjusted to the decisions made before.
The most interesting finding is that measures of domain specificity perform unexpectedly low for bigram and trigram recognition but when being applied to their unigram components they appear in the upper parts of the tree. When looking into the data, the reason for this seems to be that there is a downgrading of multi-word term candidate phrases (bigrams and trigrams) if a component (preferably the first) is too common to belong to a term. A second experiment, in which we compare term candidates with shared components confirms this finding. The components of terms are addressed in several studies (Erbs et al, 2015; Frantzi el al., 2000; Nakagawa and Mori,2003; Zhang et al., 2012), but to our knowledge this aspect of termhood has not been considered yet.
Since our model is flexible and the feature selection easily adapts to different types of text data, we plan to apply it to forum texts and see how the results differ from the ones in this study. In addition, we aim to explore whether the results are reproducible for terms from other technical domains.

References

Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. 1984. Classification and Regression Trees. Wadsworth Publishing Company, Belmont, CA.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5-32.

Kenneth W. Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 76-83, Vancouver, British Columbia, Canada.

Jonathan Cohen. 1995. Highlights: Language- and Domain-Independent Automatic Indexing Terms for Abstracting. *Journal of the Association for Information Science and Technology*, 46(3):162-174.

Merley da Silva Conrado, Thiago S. Pardo, and Solange O. Rezende. 2013. A Machine Learning Approach to Automatic Term Extraction using a Rich Feature Set. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 16-23, Atlanta, Georgia.

Nicolai Erbs, Pedro B. Santos, Torsten Zesch and Iryna Gurevych. 2015. Counting What Counts: Decompounding for Keyphrase Extraction. in *Proceedings of the ACL 2015 Workshop on Novel Computational Approaches to Keyphrase Extraction*, pages 10–17, Beijing, China.

Stefan Evert. 2005. The Statistics of Word Cooccurrences: Word Pairs and Collocations. Dissertation, Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Jody Foo and Magnus Merkel. 2010. Using Machine Learning to Perform Automatic Term Recognition. In *Proceedings of the 7th LREC - Workshop on Methods for Automatic Acquisition of Language Resources and their Evaluation Methods*, pages 49–54, Malta.

Katerina Frantzi, Sophia Ananiadou and Hideki Mima. 2000. Automatic Recognition of Multiword-Terms: the C-value/NC-value Method. *International Journal on Digital Libraries*, 3(2): 115-130.

John Justeson and Slava Katz. 1995. Technical Terminology: some Linguistic Properties and an Algorithm for Identification in Text. *Natural Language Engineering*, 1(1):9-27.

Kyo Kagueura and Bin Umino. 1996. Methods of Automatic Term Recognition: A Review. *Terminology*, 3(2): 259-289.

Mladen Karan, Jan Šnajder and Bojana D. Bašić. 2012. Evaluation of Classification Algorithms and Features for Collocation Extraction in Croatian. In *Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 657–662, Istanbul, Turkey.

Luying Liu, Jianchu Kang, Jing Yu, and Zhongliang Wang. 2005. A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering. In *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'05)*, pages 597–601, Wuhan, China.

Gunn I. Lyse and Gisle Andersen. 2012. Collocations and Statistical Analysis of n-grams: Multiword Expressions in Newspaper Text. *Exploring Newspaper Language: Using the Web to Create and Investigate a Large Corpus of Modern Norwegian, Studies in Corpus Linguistics*, pages 79–109, John Benjamins Publishing, Amsterdam, Netherlands.

Diana Maynard, Yaoyong Li and Wim Peters. 2003. NLP Techniques for Term Extraction and Ontology Population. In *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, IOS Press, Amsterdam, Netherlands.

Hiroshi Nakagawa and Tatsunori Mori. 2003. Automatic Term Recognition based on Statistics of Compound Nouns and their Components. *Terminology*, 9(2):201–219.

Pavel Pecina and Pavel Schlesinger. 2006. Combining Association Measures for Collocation Extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 651–658, Sydney, Australia.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. mwetoolkit: a Framework for Multiword Expression Identification. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Gerard Salton and Michael McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Johannes Schäfer, Ina Rösiger, Ulrich Heid and Michael Dorna. 2015. Evaluating Noise Reduction Strategies for Terminology Extraction (TIA'15). In *Proceedings of Terminology and Artificial Intelligence*, pages 123–131, Granada, Spain.

Jason Tilley. 2008. *A Comparison of Statistical Filtering Methods for Automatic Term Extraction for Domain Analysis*. Master thesis, University of Virginia.

Joachim Wermter and Udo Hahn. 2006. You Can't Beat Frequency (Unless You Use Linguistic Knowledge) - A Qualitative Evaluation of Association Measures for Collocation and Term Extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 785-792, Sydney, Australia.

Behrang Q. Zadeh and Siegfried Handschuh. 2014. The ACL RD-TEC: A Dataset for Benchmarking Terminology Extraction and Classification in Computational Linguistics. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, pages 52-63, Dublin, Ireland.

Ziqi Zhang, José Iria, Christopher Brewster and Fabio Ciravegna 2008. A Comparative Evaluation of Term Recognition Algorithms. In *Proceedings of the Sixth Conference on International Language Resources and Evaluation (LREC'08)*, pages 2108-2113, Marrakech, Morocco.

Chunxia Zhang, Zhendong Niu, Peng Jiang and Hongping Fu. 2012. Domain-Specific Term Extraction from Free Texts. *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'12)*, pages 1290–1293, Chongqing, China.

Author Index