Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

Master Thesis

Comparative Argument Mining

Mirco Franzek

5franzek@informatik.uni-hamburg.de
Studiengang Informatik
Matr.-Nr. 6781911

Erstgutachter:	Prof. Dr. Chris Biemann
Zweitgutachter:	Dr. Alexander Panchenko

Abgabe: April 2017

I am C3P0, protocol droid, human-cyborg relations.
I am fluent in over 6 million forms of communication. – *C3PO*

Contents

1 Introduction

1.1 Motivation: An Open-Domain Comparative Argumentative Machine (CAM)

1.2 Related Work

1.2.1 Argumentation Theory

[Habernal et al., 2014] presented a comparison between the results of two different annotation studies. One used the Claim/Premise-Model, while the other one used the Toulmin model. They emphasized that there is no "one-size-fits-all" model. (ERWEITERN)

(BEGRÜNDEN: warum kein modell genutzt)

1.2.2 Argument Mining

[Lippi and Torroni, 2016] gave a summary of the research topic "Argument Mining" in general. They introduced five dimensions to describe Argument Mining problems: granularity of input, the genre of input, argument model, the granularity of target and goal of analysis. Furthermore, the typical steps of Argument Mining Systems are defined. First, the input must be divided into argumentative (e.g. claim and premise) and non-argumentative parts. This step is described as a classification problem. Second, the boundaries of the argumentative units are identified; this is understood as a segmentation problem. Third, the relations between argumentative units are identified. For instance, claims and premises are connected with a "support" relation.

A system which is capable of recognising comparative sentences and their components such as the compared entities, the property on which the entities are compared to and the direction of comparison was presented in [Fiszman et al., 2007]. The evaluation showed that the outcome has a high quality (SCORE?). However, the presented system is specific to the domain of studies to drug therapy. The system uses patterns generated from sentences (WHICH SENTENCES), as well as domain knowledge. Therefore, the methods cannot be transferred for the problem of this thesis.

[Park and Blake, 2012] presented a domain-specific approach on argumentative sentence detection. The problem is formulated as a binary classification task (a sentence is either comparative or not). As in [Fiszman et al., 2007], the features are tailored for medical publications. Lexical features capture the presence of specific words, many of them bound to the medical domain. The analysis of 274 sentences resulted in syntactic features. (BEISPIEL) Similar to [Fiszman et al., 2007], the features cannot be directly transferred to other domains.

A recent publication on Comparative Argument Mining is [Gupta et al., 2017], where a set of rules for the identification of comparative sentences (and the compared entities) is derived from *Syntactic Parse Trees*. With those rules, the authors achieved a F1 score of 0.87 for the identification of comparative sentences. The rules were obtained from 50 abstracts of biomedical papers. Such being the case, they are domain dependent.

Because this thesis deals with user-generated content from the web, publications dealing with similar data are of interest.

The challenges occurring while processing texts from social media are described in [Šnajder, 2017]. In this publication, social media is broadly defined as "less controlled communication environments [...]". Besides the noisiness of text, missing argument structures and poorly formulated claims are mentioned. It is expected that the text used in this thesis will have the same shortcomings. Additionally, [Šnajder, 2017] emphasized that analyzing social media texts can delivery reasons behind opinions.

In addition to the challenges mentioned above, [Dusmanu et al., 2017] also points to the specialized jargon in user-generated content like hashtags and emoticons. With this in mind, [Dusmanu et al., 2017] classified tweets about the "Brexit" and "Grexit" either as argumentative or as non-argumentative. Besides features used in other mentioned papers, new features covering hashtags and sentiment are added. They achieved a F1 score of 0.78 (using Logistic Regression) for the classification. It must to be said that the data set is small (SIZE) and the domain is rather specific.

Many publications on argument mining are dealing with a classification problem of some kind. Publications dealing with the identification of argument structures are of relevance for this thesis, as they provide valuable insights on the suitability of features and algorithms.

[Aker et al., 2017] summarized and compared features used in other publications for identification of argumentative sentences. In addition, a Convolutional Neural Network (as described in [Kim, 2014]) was tested. Two existing corpora and six different classification algorithms were used. As a result, structural features are most expressive; Random Forest is the best classifier.

[Stab and Gurevych, 2014] described a two-step procedure to identify components of arguments (such as claim and premise) and their relationships (like "premise A supports claim B"). The identification step is formulated as a multi-class classification. For the identification of argumentative components, a F1 score of 0.72 is reported.

How different datasets represent the argumentative unit of a claim is analysed in [Daxenberger et al., 2017]. After an analysis of the datasets and their annotation scheme, [Daxenberger et al., 2017] conducted two experiments. In the first one, each learner (Logistic Regression, Convolutional Neural Networks and LSTM) was trained and evaluated (10-fold cross-validation) on each dataset one after another. On average, the macro F1 score for identifying claims was 0.67 (all results ranging from 0.60 to 0.80). No significant difference between the results of Logistic Regression and the neural models was found. In isolation, lexical, structural and word embeddings were the best features. Structural features turned out to be the weakest. The second experiment was conducted in a cross-domain fashion. For each pair of datasets, one was used as the training set and the other one as the test set. The average macro F1 score was 0.54. In this scenario, the best feature combination outperformed all neural models. However, it is assumed that there might not be enough training data for the neural models. As the last point, [Daxenberger et al., 2017] noted that all claims share at least some lexical clues.

The role of discourse markers in the identification of claims and premises are discussed in [Eckle-Kohler et al., 2015]. A discourse marker is a word or a phrase which connects discourse units (citation). For instance, the word "as" can show a relation between claim and premise: "As the students get frustrated, their performance generally does not improve". A similar function for words like "better", "worse" or "because" is expected in this thesis. [Eckle-Kohler et al., 2015] showed that discourse markers are good at discriminating claim and premises. If claim and premise are merged into one class "argumentative", this can be used to identify argumentative sentences. The F1 score is not presented, but the accuracy is between 64.53 and 72.79 percent.

A summary of several features for the identification of argumentative sentences can be found in chapter **??**.

1.2.3 Domain-Specific Comparative Systems

The enormous amount of Comparison Portals shows the need for comparisons. Frequently aired television spots empathize the popularity of those portals.

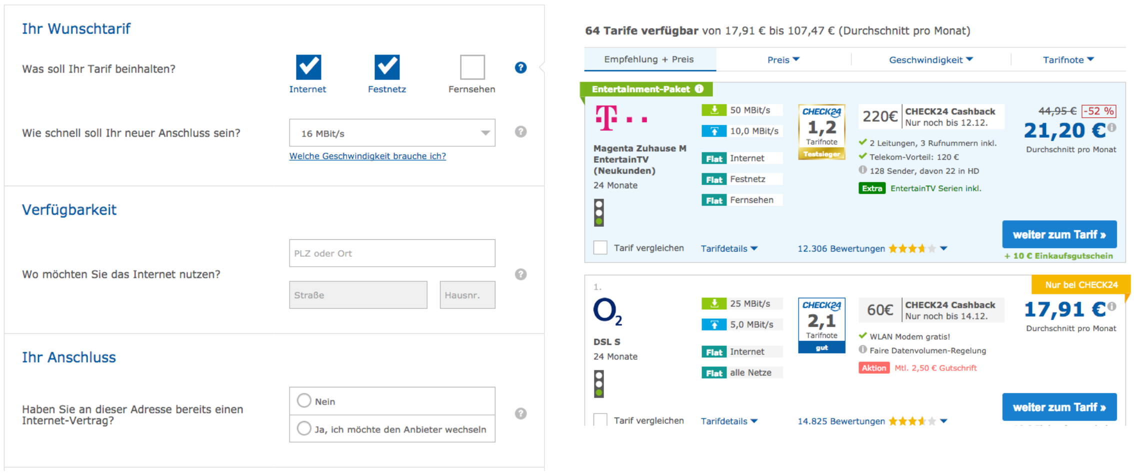
Most of those portals are specific to a few domains and a subset of properties, for example, car insurances and their price. Because of that, those systems have some restrictions. Comparisons are only possible between objects of the domains and predefined properties. Source of the data is usually databases. Humans are involved in gathering, entering and processing the data.

Comparison Portals solely compare and deliver facts. Because of that, they can only give the advice to choose X over Y based on the facts collected. An insurance X might be the best in the comparison (e.g., best price), while the internet is full of complaints about lousy service.

Examples of classical Comparative Portals are *Check24*, *Verivox*, *Idealo*, *GoCompare*, and *Compare*[1], just to name a few.

[1]https://check24.de, https://verivox.de, https://idealo.de, https://gocompare.com,

As an example, Check24 can compare a wide variety of different objects like several insurances, credit cards, energy providers, internet providers, flights, hotels and car tires. After the user entered some details (based on the object type, see figure 1.2.3), Check24 shows a ranking of different service providers. The user can choose different properties to re-rank the list. For instance, to compare different DSL providers, the user has to enter her address, how fast the internet should be and if she wants telephone and television as well. She can then select price, speed, and grade (rating) to sort the resulting list.



Ihr Wunschtarif

Was soll Ihr Tarif beinhalten? [✓ Internet] [✓ Festnetz] Fernsehen

Wie schnell soll Ihr neuer Anschluss sein? 16 MBit/s

Welche Geschwindigkeit brauche ich?

Verfügbarkeit

Wo möchten Sie das Internet nutzen? PLZ oder Ort / Straße / Hausnr.

Ihr Anschluss

Haben Sie an dieser Adresse bereits einen Internet-Vertrag? Nein / Ja, ich möchte den Anbieter wechseln

64 Tarife verfügbar von 17,91 € bis 107,47 € (Durchschnitt pro Monat)

Empfehlung + Preis | Preis | Geschwindigkeit | Tarifnote

Entertainment-Paket

Magenta Zuhause M EntertainTV (Neukunden) 24 Monate — 50 MBit/s, 10,0 MBit/s, Flat Internet, Flat Festnetz, Flat Fernsehen — CHECK24 1,2 Tarifnote Testsieger — 220€ CHECK24 Cashback Nur noch bis 12.12. — ✓ 2 Leitungen, 3 Rufnummern inkl. ✓ Telekom-Vorteil: 120 € — 128 Sender, davon 22 in HD — Extra EntertainTV Serien inkl. — ~~44,95 €~~ -52 % **21,20 €** Durchschnitt pro Monat — weiter zum Tarif » + 10 € Einkaufsgutschein

Tarif vergleichen — Tarifdetails — 12.306 Bewertungen

1. **DSL S** 24 Monate — 25 MBit/s, 5,0 MBit/s, Flat Internet, Flat alle Netze — CHECK24 2,1 Tarifnote gut — 60€ CHECK24 Cashback Nur noch bis 14.12. — ✓ WLAN Modem gratis! — Faire Datenvolumen-Regelung — Aktion Mtl. 2,50 € Gutschrift — Nur bei CHECK24 **17,91 €** Durchschnitt pro Monat — weiter zum Tarif »

Tarif vergleichen — Tarifdetails — 14.825 Bewertungen

Figure 1.2.1: Check24 DSL Provider

The other mentioned sites work similarly. They provide more of a ranking than a comparison.

Another interesting type of websites are Question Answering Portals like *Quora* or *GuteFrage*[2]. Although comparisons are not their primary goal, a lot of comparative questions are present on those sites. On Quora, more than 2.380.000 questions have the phrase “better than” in their title. If *Ruby* and *Python* are added, 10.100 questions remain.[3] Same is true for the German site GuteFrage, though, the numbers are smaller than on Quora.[4]

More interestingly are systems which can compare any objects on arbitrary properties. Two examples are *Diffen* and *Versus*[5].

Versus aggregates freely available data sources like Wikipedia and official statistic reports. For example, the comparison of “Hamburg vs. Berlin” uses Wikipedia for the number of universities, worldstadiums.com for the availability of sport facilities and the

https://compare.com - all last checked: 12.12.2017

[2]https://quora.com, https://gutefrage.net - all last checked: 12.12.2017

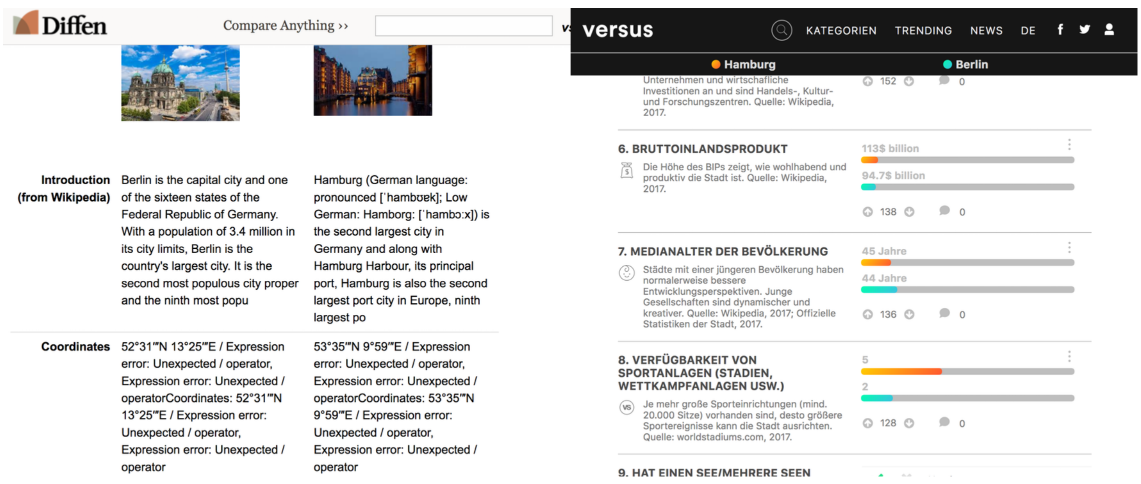
[3]Checked via Google on 11th of December. Search phrase: `"better than" site:quora.com` and `ruby python "better than" site:quora.com`

[4]334.000 for `"besser als" site:gutefrage.net` and 78 for `ruby python "Besser als" site:gutefrage.net`

[5]https://diffen.com, https://versus.com - all last checked: 12.12.2017

Economist for the Big Mac Index. Presumably, some human processing is involved as the possible comparisons are limited. For instance, a comparison of Hamburg and Darmstadt is not possible as Darmstadt is not available on Versus. Likewise, "Ruby vs. Python" is not possible, Versus suggests to compare "Rome vs. Pyongyang" instead. Although Versus shows how many users "liked" the objects, it does not give a clear statement which one is better. For instance, it is not possible to check automatically whether Hamburg or Berlin is better for a short city trip. The user must search manually all valid properties like the number of museums, theaters, the price of public transport tickets and so on.

Similar to Versus, Diffen aggregates different data sources (see figure 1.2.3). All in all, the aggregated information is similar to Versus. The comparison is also tabular. Besides the automatically aggregated data, users can add information on their own. Diffen describes itself as "inspired by Wikipedia"[6]. Diffen does not enforce any restrictions on the objects of comparison, but it faces the same problem as Versus as objects are missing. A comparison between Darmstadt and Hamburg is likewise not possible: all cells for Darmstadt in the table are just empty.



Introduction (from Wikipedia)	Berlin is the capital city and one of the sixteen states of the Federal Republic of Germany. With a population of 3.4 million in its city limits, Berlin is the country's largest city. It is the second most populous city proper and the ninth most popu	Hamburg (German language: pronounced [ˈhambʊɐ̯k]; Low German: Hamborg: [ˈhambɔːx]) is the second largest city in Germany and along with Hamburg Harbour, its principal port, Hamburg is also the second largest port city in Europe, ninth largest po
Coordinates	52°31′″N 13°25′″E / Expression error: Unexpected / operator, Expression error: Unexpected / operatorCoordinates: 52°31′″N 13°25′″E / Expression error: Unexpected / operator, Expression error: Unexpected / operator	53°35′″N 9°59′″E / Expression error: Unexpected / operator, Expression error: Unexpected / operatorCoordinates: 53°35′″N 9°59′″E / Expression error: Unexpected / operator, Expression error: Unexpected / operator

Unternehmen und wirtschafliche Investitionen an und sind Handels-, Kultur- und Forschungszentren. Quelle: Wikipedia, 2017.
152 0

6. BRUTTOINLANDSPRODUKT
Die Höhe des BIPs zeigt, wie wohlhabend und produktiv die Stadt ist. Quelle: Wikipedia, 2017.
113$ billion
94.7$ billion
138 0

7. MEDIANALTER DER BEVÖLKERUNG
Städte mit einer jüngeren Bevölkerung haben normalerweise bessere Entwicklungsperspektiven. Junge Gesellschaften sind dynamischer und kreativer. Quelle: Wikipedia, 2017; Offizielle Statistiken der Stadt, 2017.
45 Jahre
44 Jahre
136 0

8. VERFÜGBARKEIT VON SPORTANLAGEN (STADIEN, WETTKAMPFANLAGEN USW.)
Je mehr große Sporteinrichtungen (mind. 20.000 Sitze) vorhanden sind, desto größere Sportereignisse kann die Stadt ausrichten. Quelle: worldstadiums.com, 2017.
5
2
128 0

9. HAT EINEN SEE/MEHRERE SEEN

Figure 1.2.2: "Hamburg vs. Berlin" on Diffen and Versus

Neither Versus nor Diffen provides a comprehensible reason why an object is better than another one. They merely aggregate facts and bring them face to face. Despite the aggregation approach of both systems, many meaningful comparisons are not possible or not helpful (like "Hamburg vs. Darmstadt", "Java vs. C#", "Dr Pepper vs. Orange Juice"). Also, the user can not define the properties for the comparison. The sites provide every information available for the objects. For instance, Versus shows 42 properties for "Hamburg vs. Berlin" and only 35 for "Hamburg vs. Munich".

To summarize, a lot of different comparison portals exist and are widely used. Espe-

[6]https://www.diffen.com/difference/Diffen:About - Last checked: 11.12.2017

cially the domain-specific portals do a good job, but inflexibility dearly buys the performance. First, the portals can only compare objects on predefined properties. Second, the data acquisition is not fully automatic. Domain-unspecific systems are good at aggregating information but do not provide a reasonable explanation to prefer X over Y.

Adding information like comments and product reviews can enrich the comparison with reasons and opinions, such as "Ruby is easier to learn than C" or "Python is more suitable for scientific applications than Erlang as many libraries exist".

2 Building a data set for Comparative Argument Mining

Due to the novelty of Argument Mining (and especially Comparative Argument Mining), the supply of datasets is small. Thus, a new data set had to be created.

This dataset was designed to answer the questions if a given sentence compares two known objects, and if it does, if the first-mentioned object is better or worse than the second one. Those questions will be translated to several classification tasks in the later chapters.

The dataset was created using the crowdsourcing platform CrowdFlower[1]. As described in detail in the following chapters, the annotators were asked to assign one of four (and later three) classes to a sentence in which the objects of interest are highlighted.

The final dataset consists of 7421 sentences, each containing one of 273 object pairs. The sentences were labelled with one of three classes. Each sentence was at least annotated by three different annotators.

2.1 Common Crawl Text Corpus

The sentences for the crowdsourcing campaign were obtained from a CommonCrawl[2] dataset. CommonCrawl is a non-profit organisation which crawls the web and releases the crawled data for free use.

The data used in this thesis was already preprocessed[3] (see [Panchenko et al., 2017]). First, it contains only English text. Duplicates and near-duplicates were removed, as well as all HTML tags. The texts were then split into sentences.

The resulting sentences were used to obtain the sentences for the crowdsourcing campaign. To make them manageable, an ElasticSearch index (from now on called "the index") was created. The index contains 3,288,963,864 unique sentences.

To get an idea if there are enough comparative sentences in the index, it was queried for all sentences containing one of the words *“better”* or *“worse”*, as those words often indicate a comparison. This query returns 32,946,247 matching sentences. Querying for *“is better than”* still returns 428,932 sentences.

[1]https://www.crowdflower.com 23.02.2018
[2]https://commoncrawl.org 23.02.2018
[3]Download Link

Those numbers show that there are enough sentences in the index to create a dataset for the given task. Even if only 1% of the sentences containing *"is better than"* are truly comparative, there would be 4289 training examples for the machine learning algorithm.

2.2 Prestudies

Before the mainstudy could start, several questions had to be answered.

First, how to extract sentences from the index? Second, how to preprocess those sentences? Third, which labels should be assigned to the sentences? Fourth, how to phrase the guidelines?

Two prestudies were conducted to answer those questions.

2.2.1 Sentence Selection

The sentences for the crowdsourcing campaign should have a high probability of being comparative so that enough positive examples for the machine learning part are present. To ensure this, a list of cue words which indicate comparison was compiled. For the prestudy, those words were *"better"*, *"worse"*, *"inferior"*, *"superior"*, and *"because"*. Comparable objects are needed as well. A list of object pairs was selected by hand (see table 2.2.1). The pairs were selected in a way that they span a wide range of different domains, such as programming languages, countries and pets. The idea behind this is that pets are compared differently than programming languages. In this way, there will be different comparison patterns in the data.

Table 2.2.1: Objects of the Annotation Prestudy

First Object	Second Object	# Sentences
Ruby	Python	100
BMW	Mercedes	100
USA	Europe	100
Beef	Chicken	100
Android	iPhone	100
Cat	Dog	100
Football	Baseball	100
Wine	Beer	100
Car	Bicycle	100
Summer	Winter	100

However, not all comparisons will contain one of the cue words mentioned above. Two different queries were used to overcome the coverage problem. Sevenhundred-fivtey sentences were obtained using query 2.1 (seventy-five for each pair) and 250 using query 2.2 (twenty-five for each pair). The second query will also match not-anticipated sentences such as *"I like X more than Y since Z."*.

Listing 2.1: Prestudy Sentence Selection Query A

```
{ "query":{  "bool":{ "must":[ {
          "query_string":{
            "default_field":"text",
            "query":"(better OR worse OR superior OR inferior) AND \"<
               ↪ OBJECT_A>\" AND \"<OBJECT_B>\""
          }
        } ] } } }
```

Listing 2.2: Prestudy Sentence Selection Query B (shortened)

```
[...]
          "query_string":{
            "default_field":"text",
            "query":" \"<OBJECT_A>\" AND \"<OBJECT_B>\""
[...]
```

Table 2.2.2 shows some sentences obtained with this method. The objects of interest ar printed in italics.

Table 2.2.2: Extracted Sentences

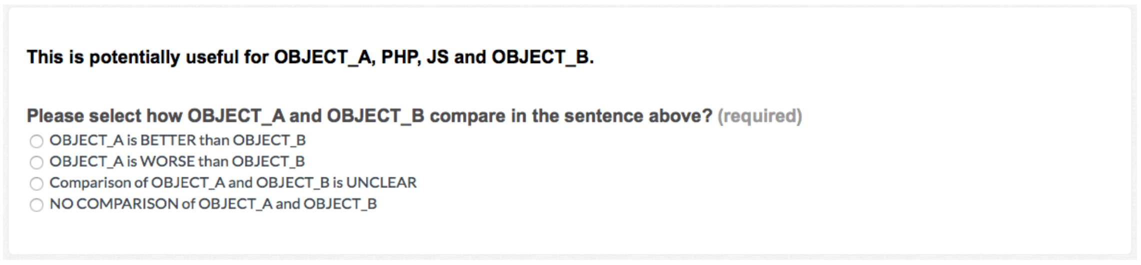
Sentence	Cue Words Used
He's the best pet that you can get, Better than a *dog* or *cat*.	Yes
Android phones have better processing power than *iPhone*	Yes
10 Things *Android* Does Better Than *iPhone* OS	Yes
Dog scared of *cat*	No
In fact, many 'supercars' will use *BMW* or *Mercedes* engines.	No

2.2.2 Prestudy A

The first prestudy had two goals. First, it should assess if the sentence selection method returns enough comparative sentences. Second, the design of the study as described below should be checked. On that account, a crowdsourcing campaign with one hundred of the 1000 sentences was started.

For each sentence, the annotators should decide to which class a sentence belongs. The classes are described in table 2.2.3. The classes `BETTER`, `WORSE` and `NO_COMP` directly refer to the problem stated at the beginning of this chapter. The class `UNCLEAR` was added to capture all sentences which are somehow comparative but do not fit into the classes `BETTER` or `WORSE`.

In each sentence, the first object of interest was replaced with `OBJECT_A`, while the second one was replaced with `OBJECT_B`. Table 2.2.5 shows examples of processed sentences. The idea behind this was to enable the annotators to quickly see which objects should be taken into account for assigning a class. For example, in sentence three of table



This is potentially useful for OBJECT_A, PHP, JS and OBJECT_B.

Please select how OBJECT_A and OBJECT_B compare in the sentence above? (required)

- ○ OBJECT_A is BETTER than OBJECT_B
- ○ OBJECT_A is WORSE than OBJECT_B
- ○ Comparison of OBJECT_A and OBJECT_B is UNCLEAR
- ○ NO COMPARISON of OBJECT_A and OBJECT_B

Figure 2.2.1: Annotator view (Prestudy A)

Table 2.2.3: Classes for Prestudy A and B

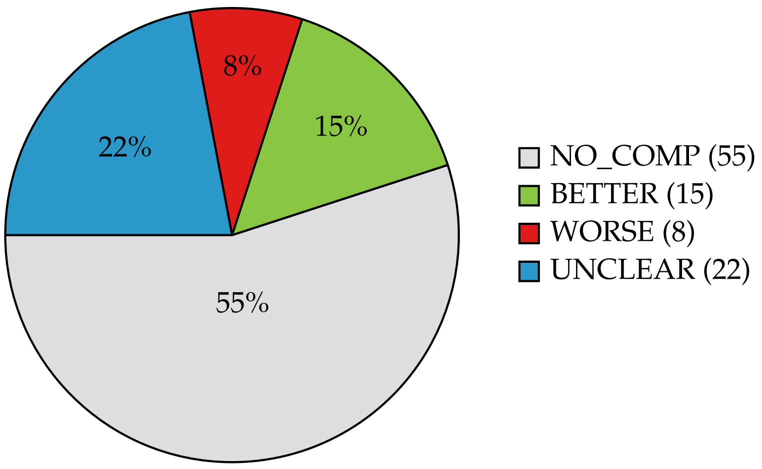
Class	Description
BETTER	The first object in the sentence (object A) is better than the second one (object B)
WORSE	The first object is worse
UNCLEAR	Neither BETTER nor WORSE fits, but the sentence is comparative
NO_COMP	The sentence is not comparative or the sentence is a question

2.2.5, the annotator might be confused which of the objects are of interest, yet the replacement makes it clear that he should ignore *C* and *VB*. The view of the annotator (for a single sentence) is shown in figure 2.2.2.

Per batch, each annotator saw five sentences to annotate. He was also able to look into the annotation guidelines anytime he wanted. To filter out low-quality annotators, twelve sentences were selected as test questions. Each participant took a quiz (eight test questions) before the actual annotation process. One of the five sentences of the batch was a test question as well. If the annotator missed more than 30% of the test questions, he was removed from the task.

Figure 2.2.2 shows the resulting class distribution. The numbers after the class names show the absolute members of that class. As 45% of the sentences are comparative, the selection procedure works satisfying.

Figure 2.2.2: Class Distribution (Prestudy A)



The agreement of the annotators was acceptable. For thirty-seven sentences, all annotators agreed on one classes. Only for five sentences, each annotator assigned a different class. The remaining fifty-eight sentences got two different classes.

Table 2.2.4: Uncertain Sentences (Prestudy A)

#	Sentence	Ann. 1	Ann. 2	Ann. 3
1	The only reason OBJECT_A is used over OBJECT_B, is because of libraries...	WORSE	UNCLEAR	NO_COMP
2	Agile development is the most popular model at the moment because of architectures like OBJECT_A on Rails and Django (for OBJECT_B)	NO_COMP	NO_COMP	UNCLEAR
3	Your C and VB devs can suddenly easily write web apps and your OBJECT_A and OBJECT_B devs can too - with the added bonus of much better performance	NO_COMP	NO_COMP	UNCLEAR
4	I'm a huge OBJECT_A/Django & OBJECT_B/Rails fan, but I will never stop using PHP because it is so broadly accepted and supported	NO_COMP	UNCLEAR	UNCLEAR
5	It's why I mention OBJECT_A and OBJECT_B, because they've at least heard of them.	NO_COMP	NO_COMP	UNCLEAR

Some uncertain sentences are shown in table 2.2.5, which displays the sentence and the decision of each annotator. As one can see in sentence two to five, annotators frequently were not able to distinguish between `NO_COMP` and `UNCLEAR`. This is the case in fifty-one of the fifty-eight cases were two different classes were assigned.

Fourteen out of fifty-five participants took part in an exit survey to rate the task. The overall satisfaction was rated with 3.2 out of 5. While the instructions (4.5), difficulty (4.4) and payment (3.8) got acceptable to good ratings, the test questions (2.9) were critizied. Also, 32 potential annotators failed the quiz. A second prestudy was conducted to adress the discovered problems.

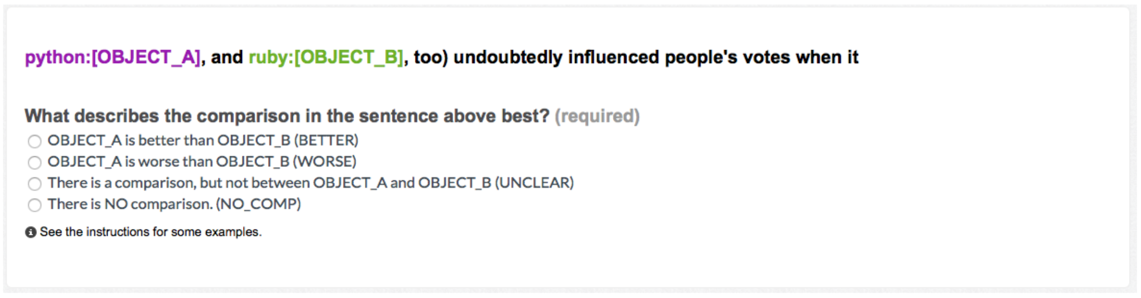
2.2.3 Prestudy B

In the second prestudy, 200 sentences were annotated. Some changes in the task design were made to address the shortcomings of the first study.

Some points were identical to the first study. As the sentence selection process worked fine, the same 1000 base sentences were used in the second prestudy. Each sentence was annotated by three annotators. They had to pass a quiz of eight test questions and had to keep an accuracy of 70% on the test questions during the annotation procedure. The classes were the same as well.

The title of the task now contained the information that some computer knowledge is required for this task, as some pairs come from this domain. To address the problem with the confusion between `UNCLEAR` and `NO_COMP`, the wording on this classes was changed. The new view of the annotator is displayed in figure 2.2.3. In the first prestudy, some annotators complained that the test questions were not fair. In fact, they contained some special cases so that they did not represent the whole dataset in an appropriate way. In the second prestudy, fifty-one test questions were used, who cover a wider range of examples.

The sentence preprocessing was altered as well. Instead of replacing the object, **:[OBJECT_A]** or **:[OBJECT_B]** was appended. The colon and square brackets emphasize where the object of interest ends and the suffix begins. The idea behind this was, that the removal of the objects also removes some context from the sentences, which might be needed to classify them correctly. For further highlighting, the objects had a different colour.


python:[OBJECT_A], and ruby:[OBJECT_B], too) undoubtedly influenced people's votes when it

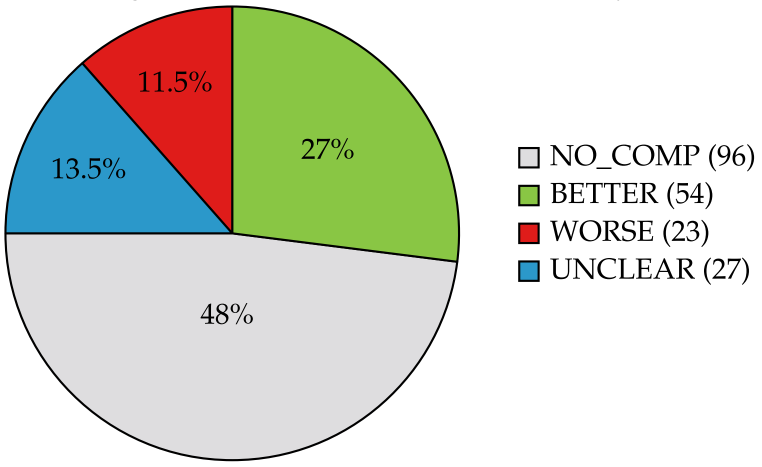
What describes the comparison in the sentence above best? (required)
- ◯ OBJECT_A is better than OBJECT_B (BETTER)
- ◯ OBJECT_A is worse than OBJECT_B (WORSE)
- ◯ There is a comparison, but not between OBJECT_A and OBJECT_B (UNCLEAR)
- ◯ There is NO comparison. (NO_COMP)

See the instructions for some examples.

Figure 2.2.3: Annotator view (Prestudy B)

The resulting class distribution is presentend in figure 2.2.4.

Figure 2.2.4: Class Distribution (Prestudy B)



As in the first prestudy, nearly half of the sentences are somewhat comparative. For

125 (62.5%) sentences, all annotators agreed on one class. Four (2.0%) sentences got three different classes, for the remaining seventy-one (35.5%) sentences two of the three annotators agreed on one class. The confusion between `UNCLEAR` and `NO_COMP` is still the main problem for the sentences where only two annotators could agree. However, in the second prestudy this confusion only makes up fourty-five out of the seventy-two cases (62.5% instead of 87.9% in the first prestudy). Compared to the first prestudy, the amount of sentences where all annotators agreed on one class increased from 37% to 62.5%.

Table 2.2.5: Uncertain Sentences (Prestudy B)

#	Sentence	Ann. 1	Ann. 2	Ann. 3
6	Google shouldn't have mandated an inferior map app on the iphone:[OBJECT_A] (as opposed to android:[OBJECT_B]).	BETTER	WORSE	NO_COMP
7	(See android:[OBJECT_A] Dethrones the iphone:[OBJECT_B] .)	BETTER	NO_COMP	NO_COMP
8	android:[OBJECT_A] didn't out pace the iphone:[OBJECT_B] this year, it just sold slightly better in America	WORSE	NO_COMP	WORSE
9	To me it is much better than iphone:[OBJECT_A] and android:[OBJECT_B].	NO_COMP	UNCLEAR	UNCLEAR
10	(android:[OBJECT_A] , Crush , iPad , iphone:[OBJECT_B])	NO_COMP	BETTER	NO_COMP

From 125 candidate annotators, thirty-five failed the initial quiz. Twelve annotators were removed during the annotation process as they answered too many test questions wrong. Twenty-two annotators took the exit survey. The overall satisfaction increased to 3.7 out of 5. The test question fairness was now rated with 3.7 out of 5 instead of 2.9. The rating for the payment slightly increased to 3.9, yet the payment was not changed. However, the rating for the instructions decreased to 3.9 and for the difficulty to 3.5. The change in numbers is explained by the increased amount of sentences, which introduce new cases which are not directly reflected in the annotation guidelines. However, as only a small fraction of the annotators took the exit survey in both prestudies, the results can only be used as one signal. The annotation results are another, more important signal, and they are convincing.

2.2.4 Validation of results

Due to an error in the creation of the crowdsourcing task, the sentences where not shuffled. This means that the first one-hundred sentences of the second prestudy are the same as the one-hundred sentences of the first prestudy. Another problem is the bias: except for X sentences, all other sentences contained the pairs X and Y. However, since the goal of the prestudy was mainly to assess the sentence selection method and the guidelines, this does not invalidate the results. This problems were removed in the main study.

2.3 Main Study

2.3.1 Task Description

2.3.2 Data Generation

Three domains were fixed for the sentences of the main study. The domains were chosen in a way that a majority of people can decide whether a sentence contains a comparison or not.

The most specific domain was "Computer Science Concepts". It contains objects like programming languages, database products and technology standards such as Bluetooth and Ethernet. Many computer science concepts can be compared objectively, for instance, one can compare Bluetooth and Ethernet on their transmission speed. Some basic knowledge of computer science was needed to label sentences correctly. For example, to compare Eclipse and NetBeans, the annotator must know what an Integrated Development Environment (IDE) is and that both objects are Java IDEs. The need of the knowledge was communicated to the prospective annotators. The objects for this domain were manually extracted from "List of ..." articles from Wikipedia.

The second, broader domain was "Brands". It contains objects from of different types (car brands, electronics brands, and food). As brands are present in everyday life of people, it is expected that anyone can label the majority of sentences containing well known brands such as Coca-Cola or Mercedes. As with computer science, the objects for this domain were extracted from "List of ..." articles from Wikipedia.

The last domain is not restricted to any topic. For each one of 25 randomly selected seed words, ten similar words were extracted using JoBimText, a software package for distributional semantics. The seed words were created using https://randomlists.com[4]. Listing 2.3 shows the result[5] for the seed word *harvard*.

Listing 2.3: Similar words to "Harvard"

```
{
   "results":[
      { "score":688.0, "key":"harvard#NP" },
      { "score":245.0, "key":"yale#NP" },
      { "score":163.0, "key":"princeton#NP" },
      { "score":152.0, "key":"mit#NP" },
      { "score":143.0, "key":"stanford#NP" },
      { "score":133.0, "key":"university#NP"},
      { "score":132.0, "key":"tufts#NP" },
      { "score":130.0,"key":"cornell#NP"},
      { "score":127.0, "key":"nyu#NP" },
```

[4]Last checked: 25.01.2018

[5]http://ltmaggie.informatik.uni-hamburg.de/jobimviz/ws/api/stanford/jo/similar/harvard%23NP?numberOfEntries=10fo
Last checked: 25.01.2018; Some uninteresting fields were removed for brevity

```
        { "score":113.0, "key":"university#NN" }
    ]
}
```

This method covers a wide are of possible comparison patterns.

Especially for brands and computer science, the object lists are long (4493 brands and 1339 for computer science).The frequency of each object was checked using a frequency dictionary to reduce the number of possible pairs. All objects with a frequency of zero and ambiguous objects were removed from the list. For instance, the objects "RAID" (a hardware concept) and "Unity" (a game engine) were removed from the computer science list as they are also regularly used nouns.

The remaining objects were combined to pairs. For each type, all possible combinations were created. For brands and computer science, the type is the source list. For the unrestricted domain, the seed word was used. This procedure guarantees that only meaningful pairs are created. The ElasticSearch Index was then queried for entries containing both objects of each pair. For 90% of the queries, the marker terms where added to the query. This was done to check whether there is a chance that those two objects were compared. All pairs were the query yielded at least 100 sentences were kept. Those pairs are frequent enough and have a high chance of generating comparative sentences.

From the sentences of those pairs, 2500 for each category were randomly sampled as candidates for the crowdsourcing campaign. 250 sentences were manually labelled to check if there are enough comparative sentences. Those labels were discarded for the crowdsourcing campaign. The label distribution of the 250 sentences is presented in the figures FIGURE NUMBERS.

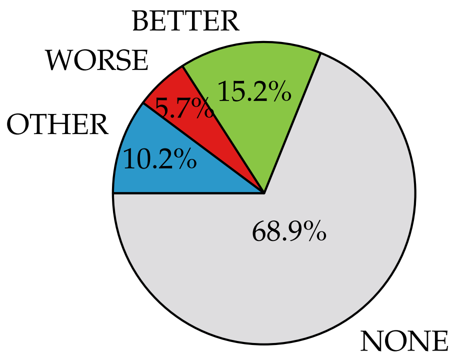


Figure 2.3.1: Brands

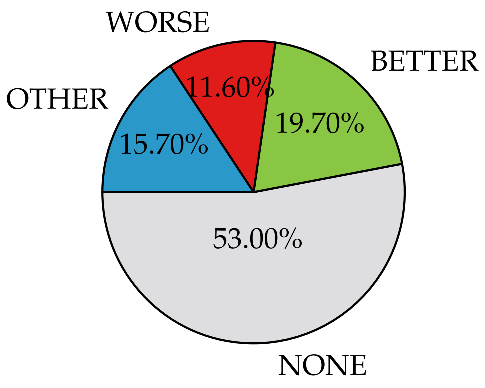


Figure 2.3.2: Computer Science

In all samples, at least 30% of the sentences are comparative. This number shows that the sampling method is sufficient to sample sentences for the crowdsourcing campaign.

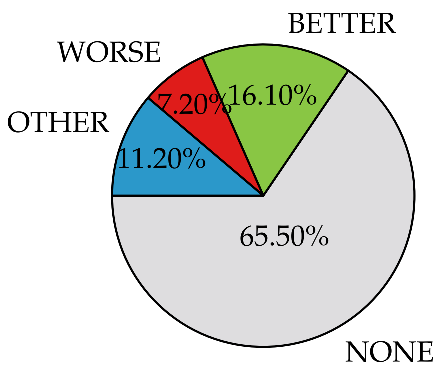


Figure 2.3.3: Unrestricted

3 Classification of Comparative Sentences

3.1 Description of the experiments

The goal of this thesis is to train a machine learning model in a way that it can decide whether a sentence contains a comparison between two defined objects or not.

To achieve this goal, the task was formulated in three ways of different granularity. In all tasks, the machine learning model was provided with the sentence and the objects.

In the first experiment, the model was trained to classify a sentence into one of the four categories described above. As stated below, the class OTHER adds uncertainty to the model. This problem is handled in next experiments. In the second experiment, all sentences with this class were removed prior training. In the third experiment, the class OTHER was joined with the class NONE. Thus, in those experiments, the model has to decide between three classes.

In the fourth experiment, BETTER, WORSE and OTHER are joined to the class ARG while the fifth experiment joins BETTER and WORSE to ARG and OTHER with NONE. Thus, the fourth and fifth experiments are binary classification tasks.

3.2 Evaluation

The evaluation of the results was done with stratified k-fold cross-validation where k is 3. The overall F1 score of each fold is the weighted average of the F1 scores of each class, where the weights are the number of examples per class. It was produced using the `classification_report` function of sklearn. While discussing the features, single F1 scores are presented. Those scores are the unweighted average of the F1 scores of each fold.

Following [Daxenberger et al., 2017], the results were also evaluated with training on two of the domains and evaluating on the leftover one.

3.3 Algorithms

The classification was performed with SkLearn ([Pedregosa et al., 2011]).

3.4 Features

Several features and feature combinations were tested. The results are presented in table X and Y. As described above, the F1 scores in the tables are the unweighted averages of the three F1 scores of each fold.

Every feature was tested on different parts of the sentence: the whole sentence, all words before the first object, all words after the second object and all words between objects. In doing so, the objects either stayed in the sentence, were removed or replaced. Two different replacement strategies were tested: replacing both objects with OBJECT and replacing the first object with OBJECT_A and the second object with OBJECT_B. The replacement approach should test if the objects influence the decision of the classifier. For example, if "Python" is always the "better" object the classifier might become biased.

The following section only describes features which worked out well, leaving out bad combinations (for example, using only the first or last part was not helpful in most cases).

In the first step, n-gram models where tested. Every uni-, bi and trigram in the whole training set was implemented as a binary feature. Restricting on frequency was not helpful. Also, trigrams were not helpful which can be explained by the length of the sentences.

3.5 Results

3.5.1 Three labels

3.5.2 Binary

3.6 Discussion

4 Conclusion

Bibliography

[Aker et al., 2017] Aker, A., Sliwa, A., Ma, Y., Lui, R., Borad, N., Ziyaei, S., and Ghobadi, M. (2017). What works and what does not: Classifier and feature analysis for argument mining. In *Proceedings of the 4th Workshop on Argument Mining*, pages 91–96.

[Daxenberger et al., 2017] Daxenberger, J., Eger, S., Habernal, I., Stab, C., and Gurevych, I. (2017). What is the essence of a claim? cross-domain claim identification. *CoRR*, abs/1704.07203.

[Dusmanu et al., 2017] Dusmanu, M., Cabrio, E., and Villata, S. (2017). Argument mining on twitter: Arguments, facts and sources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2307–2312.

[Eckle-Kohler et al., 2015] Eckle-Kohler, J., Kluge, R., and Gurevych, I. (2015). On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *EMNLP*, pages 2236–2242.

[Fiszman et al., 2007] Fiszman, M., Demner-Fushman, D., Lang, F. M., Goetz, P., and Rindflesch, T. C. (2007). Interpreting comparative constructions in biomedical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 137–144. Association for Computational Linguistics.

[Gupta et al., 2017] Gupta, S., Mahmood, A. A., Ross, K., Wu, C., and Vijay-Shanker, K. (2017). Identifying comparative structures in biomedical text. *BioNLP 2017*, pages 206–215.

[Habernal et al., 2014] Habernal, I., Eckle-Kohler, J., and Gurevych, I. (2014). Argumentation mining on the web from information seeking perspective. In *ArgNLP*.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

[Lippi and Torroni, 2016] Lippi, M. and Torroni, P. (2016). Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2):10:1–10:25.

[Panchenko et al., 2017] Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S. P., and Biemann, C. (2017). Building a web-scale dependency-parsed corpus from commoncrawl.

[Park and Blake, 2012] Park, D. H. and Blake, C. (2012). Identifying comparative claim sentences in full-text scientific articles. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 1–9. Association for Computational Linguistics.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Šnajder, 2017] Šnajder, J. (2017). Social media argumentation mining: The quest for deliberateness in raucousness.

[Stab and Gurevych, 2014] Stab, C. and Gurevych, I. (2014). Identifying argumentative discourse structures in persuasive essays. In *EMNLP*, pages 46–56.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.
Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den