News for the Advanced: A Simple Approach to Automatically Detecting Satire

Matthijs ten Wolde
6279945

Abstract

Figurative language is seen as the topic that pains Natural Language Processing (NLP) the most. The intention of the figurative language goes beyond the literal meaning of the words. This makes it a real challenge for both humans and computers. Automatic implementations for irony and sarcasm detection are widely researched in English. Satire is a form of figurative language that tries to imitate real news, exposing individuals or organisations to ridicule. Earlier work suggests that an automated assistive tool that detects satire could be the first step in fighting the growing world of fake news. This paper introduces a simple machine learning classifier to automatically detect satire in Dutch headlines, using general features based on textual markers and sentiment. The results of our classifier are promising for further research on automatic figurative language detection in Dutch, but there is some work to do there. Some suggestions for future research are also included

16/04/2021
bachelor Kunstmatige Intelligentie, UU
Supervisor: *Rick Nouwen*
Second Examiner: *Anna Wegmann*
7,5 EC

1. Introduction

The size of data created on the internet everyday was estimated to be around 2.5 Exabytes (2.5 billion gigabytes) in 2016[8]. These massive amounts of data have caused a rapid growth in the demand for text analysis techniques. These techniques extract information from textual data, like social media posts, online forums, news articles etc[9]. Sentiment analysis is a text analysis method which can give an insight into the general beliefs or feelings people have with a topic, like a company product or the policy of a political party.
Figurative language is seen as the biggest challenge for present-day sentiment analysis methods and many other Natural Language Processing (NLP) applications[6]. This type of language is used to communicate a more complex meaning than the literal meaning of the words, by making use of devices such as metaphor, sarcasm, irony, satire, analogy, and so on[14]. While humans struggle with the complex underlying meaning of figurative language themselves, it is even harder for computers[14]. Figurative language devices, like sarcasm or satire, make use of cognitive abilities of the reader to represent meaning beyond the literal. If these abilities are not present or inadequate, the real meaning is not understood and the figurative effect is lost[6].
Satire is a figurative device which deliberately exposes or ridicules individuals or organisations[3]. It is very often compared to irony and sarcasm[6]. Many cases of satire require some sort of world-knowledge about the subject in the sentence[3], which are difficult to represent in a computational way[6]. Consider the following sentences:
1. *Balen! Wij weten wie de vriendin van Rutte is, maar we zijn een kwaliteitsmedium.*
2. *Samsung presenteert nieuw bomsuit om Galaxy Note 7 te ontmantelen.*

Most Dutch-speaking people interpret these sentences as satire. This is because they possess the extra information that is needed to understand them. For sentence 1, this extra information consists of knowledge about Mark Rutte and gossip magazines. Sentence 2 can be successfully interpreted with the knowledge about the malfunctioning of the Galaxy Note 7. This information is not available for computers, which is why satire is so challenging. Irony, sarcasm and satire are inevitably a part of informal online conversations, so a system that could successively interpret this figurative language would be a big outcome for sentiment analysis and other Natural Language Processing applications[5].
Automatic irony and sarcasm detection are widely researched since the earliest known work on this topic was published by Tepperman et al. (2006)[1]. Joshi et al. (2017) summarise past research on automatic sarcasm detection. A large proportion of the work focused on classifying tweets, as Twitter makes it really easy to collect and filter tweets (using hashtags like #sarcasm). Others use longer sentences, like movie/book reviews or news articles, all labeled sarcastic or not sarcastic. Various classifiers are tested and compared using different combinations of features, like frequency features, sentence length, capitalization, semantic similarity of words, sentiment etc.
Most of the work is done on English irony, but Liebrecht et al. (2013)[4] trained a machine learning algorithm on 78 thousand sarcastic Dutch Tweets. This algorithm was tested on 3.3 million Tweets, containing 135 Tweets marked with a sarcasm hashtag. 101 of the 135 sarcastic tweets were correctly classified as sarcasm. This same classifier was tested on 250 Tweets that were given the highest confidence-value regarding the sarcasm label (the Tweets were also manually checked as being sarcastic). Using this set of tweets, the average precision of the algorithm was only 30%. Liebrecht et al. concluded that it is very hard to detect sarcasm in such an open setting.

Burfoot & Baldwin (2009) investigated whether it is possible to detect satire on the basis of simple lexical features, without having the extra information that humans normally would have[3].
This paper will try to answer the same question that Burfoot & Baldwin researched, but for the Dutch language. *To what extent can we detect satire in Dutch news headlines using simple textual features, without the related world-knowledge?* This could result in a simple approach for

automatically detecting figurative language in text. If such a simple approach does not work, we would need to give the computer access to the extra information that is needed to understand the satire. This would be a much more complex and inefficient solution.
The research in this paper will use simple lexical features extracted from Dutch headlines. With these features, a simple machine learning algorithm will be trained and tested. During this process, the relevance of features will also be tested, as some features could be more relevant for detecting satire than others.

The following sections will be structured as follows. Section 2 will describe past research on automatic detection of irony, sarcasm and satire. The data that is used and how this data is processed to be usable for automatic detection is described in Sections 3.1 and 3.2. The actual experiment and it's results are presented in Section 3.3. Then, in Section 4, we conclude our research and analyse the results. We end with a list of references and an Appendix that contains additional information about the used data and a link to the Python code written for this research.

2. Related Research

The relevance for research in automatic satire detection for many NLP applications is clear, but Rubin et al. (2016) address another application of this kind of research in their paper[17]. The high rate of media consumption and declining trust in news institutions cause for the spreading of intentionally or unintentionally misleading information. While satire normally includes cues that reveal that it should be interpreted as fake, it still misleads readers who are unaware of the underlying satirical intentions. On top of that, readers could be lacking the cultural background or contextual knowledge that is needed to correctly interpret satire. A report from 2014 shows that only 60% of Americans read beyond the headline[18]. These readers could benefit from an automated assistive tool that flags or filters inaccurate or false content. Rubin et al. hope to contribute to such a tool by building a model to distinguish between satire and real news, using features based on absurdity, humour, grammar, negative affect and punctuation. Classifying these using a Support Vector Machine[12] resulted in an F-score of 0.87. This research could help to minimize the deceptive impact of satirical news, and possibly even (intentionally) fake news[17].

Many papers have tried to give a formal definition of figurative devices like irony and satire, but this seems to be very difficult[1]. Reyes et al. (2011) show different views on the similarities and differences of these figurative language devices. It is widely considered that irony has two types: verbal irony and situational irony[6]. With verbal irony, the opposite of the literal meaning is intended. Situational irony does not imply the opposite meaning, but involves an unexpected or absurd attribute in a situation. Sarcasm is usually seen as a form of verbal irony[1]. Also, satire is often compared with irony[6]. In another paper, Reyes et al. (2012) state that irony, sarcasm and satire are overlapping figurative phenomena, which only differ in tone, usage and obviousness[13]. For many people, there is not a clear distinction between irony and sarcasm, or irony and satire[14]. Furthermore, they share more similarities than differences. Reyes & Rosso assumed sarcasm, satire and other figurative devices (e.g. hyperbole, rhetorical questions & jocularity) to be extensions of the general concept of irony[14]. Satire is traditionally divided into two options: Juvenalian, which is seen as more aggressive and pessimistic, and Horatian, the more playful and teasing style[17]. Both styles are a mixture of laughter and scorn. It is also said that satire should have a purpose, instead of just mocking a target[17]. This purpose could be to criticize someone/something, or to call for action.

The automatic detection of figurative language devices in text from social media has been very popular in recent years[4]. Twitter is the biggest source of data[1]. Reyes et al. (2011) collected a corpus of 50000 Tweets, with 10000 ironic Tweets searched using the hashtag *#irony*[6]. The Tweets were processed as features regarding ambiguity, polarity, unexpectedness and emotional scenarios. Reyes et al. (2013) used the same approach, using hashtags like *#irony* and *#politics* to collect 40000 tweets[13]. The feature set is similar to the one by Reyes et al. (2011), with features concerning textual markers, unexpectedness, style and emotional scenarios. Burfoot & Baldwin (2009) used another source of data for their research[3]. They focused on classifying English news articles as satire or true news. They used a bag-of-words model, which uses the frequency of occurrences of each word in a sentence as a feature, and features capturing slang, profanity and absurdity in satire[3]. To implement the absurdity feature, named entities (e.g. persons or organisations) in a sentence are checked to occur in the same combination on the web. This will capture absurd combinations of named entities that will likely be present in satire. Other than social media text and news articles, product reviews are also used. This does need some more work, as the ironic and unironic reviews have to be manually separated. Reyes & Rosso (2014) compare the results of an irony classifier on movie reviews, book reviews and satirical articles[14]. Their feature set is based on textual markers, emotional scenarios and unexpectedness. Buschmeier et al. (2014) collected 1254 Amazon reviews, by searching for pairs of reviews on the same product so that one of the reviews is ironic and the other is not[15]. In addition to some general features, they took in account meta-data that came with the amazon reviews, which was the rating that the product received with the review. This star-rating turned out to be a very strong indicator of irony when it was used as a feature.
Many different classifiers have shown good results in the automatic detection of irony, sarcasm and satire. Reyes et al. (2011) used a decision tree classifier on different feature combinations, achieving F-scores from 0.56 to 0.90[6]. This type of algorithm walks through a decision tree and decides, according to features, which path to take at every node. Reyes et al. (2013) also use a decision tree algorithm, and compare this to a Naïve Bayes classifier[13]. The latter uses features independently from each other to calculate the probability that some data belongs to one of the classes. The decision tree algorithm scored slightly better than the Naïve Bayes classifier, with an F-score of 0.76 compared to 0.73. The automatic satire detection by Burfoot & Baldwin (2009) was done with a Support Vector Machine[3], a simple and widely used machine learning algorithm. Their best result showed an F-score of 0.798. Hernández Farías et al. (2015) show results of 6 different classifiers widely used in text classification tasks, with the Support Vector Machine having the best performance, achieving an F-score of 0.80[2]. Lastly, the Support Vector Machine of Buschmeier et al. (2014) showed an F-score of 0.72[15].

All the papers we discussed differ slightly in the used data, the feature sets and the classifiers, but the structure that is used in the automatic detection methods for the different figurative language forms is mostly the same, hence the results could be relevant for all figurative language devices in Dutch. Burfoot & Baldwin (2009) state that the automatic satire detection task is similar to spam filtering and sentiment classification in two ways: It is a binary classification task and it is an intrinsically semantic task, meaning that satire can be detected with world knowledge about the subjects in question[3].

Our research is mostly inspired by the work of Reyes et al. (2013)[13] and Hernández Farías et al. (2015)[2]. The irony model of Reyes et al. is based on four types of features: signatures, unexpectedness, style and emotional scenarios. The features are selected to represent properties of irony as found in the literature. The rest of this section will further go into these features and the feature set of Hernández Farías et al.

Signatures

These features are focused on capturing irony in the form of textual markers or signatures, such as punctuation marks, emoticons and terms that suggest an opposition in text. In irony, quotes or capitalization are often used to guide attention to certain aspects of the text[13]. Furthermore, adverbs can hint at opposition in text. This feature set consists of three different features: pointedness, counter-factuality and temporal compression. *Pointedness* tries to target explicit marks which reflect a sharp distinction in the message that is communicated. This feature considers punctuation marks (such as ., …, ;, ?, !), emoticons, quotes and capitalization. *Counter-factuality* is focused on implicit marks. Terms that hint at opposition or contradiction in text, like 'nevertheless' or 'yet' are checked to occur. Also, adverbs that occur in negations and their synonyms are represented in this feature. Finally, *Temporal compression* identifies elements that relate to opposition in time, like an abrupt change in text. A set of temporal adverbs is used, including 'suddenly', 'abruptly' or 'now'.

Unexpectedness

Irony uses unexpectedness and incongruity to make sure that ironic text is not taken literally[13]. This unexpectedness is represented with two features: temporal and contextual imbalance. *Temporal imbalance* targets the degree of opposition between words in present and past tense in a text. Other than the temporal compression feature discussed earlier, this feature focuses on oppositions relating verbs only. *Contextual imbalance* is used to capture inconsistencies in the context. This is done with a lexical database which calculates the semantic similarity of pairs of words in a text. A semantic relatedness score is then computed for a sentence by summing the scores and dividing by the length of the text. A high contextual imbalance (suggesting ironic text) is then represented as a low semantic similarity, and vice versa.

Style

This set of features focuses on stylistic elements that are possibly recurring in irony. This style feature set is represented by three features: character n-grams, skip-grams and polarity s-grams. *Character n-grams (c-grams)* captures frequent series of morphological information, like affixes or suffices (e.g. -ty). Series of 3-5 characters are considered. This features tries to detect the irony used in sentences like "*Infan*ts don't enjoy *infan*cy like *adult*s do *adult*ery", where affixes/suffixes give an ironic effect. In *Skip-grams (s-grams)*, complete words are targeted. This feature tries to capture sequences of words that are common for irony. Instead of looking for adjacent words like normal n-grams, the skip-grams look for word sequences with arbitrary gaps. A sentence like "There are too many crazy people in my psychology class" contains the 2-gram "too many", but this feature is interested in, for example, "too crazy", the 2-sgram with a 1 token gap. Finally, *Polarity s-grams (ps-grams)* produce sequences of positive and negative terms in a sentence. This feature is relevant because positive terms are generally used to express a negative meaning in irony[13]. An open source sentiment analysis library is used to tag the s-grams with positive or negative tags. As an example, the sentence "I need more luck. I need Jesus and I'm an atheist…" is assigned the following polarity sequence: $positive_{need}$ - $positive_{jesus}$ - $negative_{atheist}$. This example sequence indicates the presence of irony because of the switch in polarity.

Emotional scenarios

Emotional scenarios can represent information that goes beyond grammar. For example, ironic expressions on social media use emoticons to safely communicate the intended ironic effect[13]. This feature set attempts to capture characteristics of irony regarding sentiment, feelings and moods. The set consists of the following features: activation, imagery and pleasantness. *Activation* stands for a degree of response in an emotional state, being either active or passive. A term like 'burning' is seen as more active than 'basic'. Then, the *Imagery* of a word tells us how easy or difficult it is to form a mental picture of that word. 'Never' is very difficult to represent mentally, while 'alcoholic' is

easier. *Pleasantness* describes a degree of pleasure that is paired with a word (e.g. 'love' is more pleasant than 'money'). All the features in this feature set are made with a dictionary that links over 8000 English words to the corresponding emotional scenario scores.

Later, Hernández Farías et al. (2015) build a feature set by combining features by Reyes et al. with some of their own[2]. They add Part of Speech (POS) features and a Sentiment Score feature, and use them together with the features from Reyes et al., except for the style features. The Part of Speech features are processed with a POS-tagger, which labels every word in a sentence with their part of speech to take into account the frequency of nouns, verbs, adjectives, etc. This is done to capture certain grammatical properties that are recurring in irony. The sentiment score feature expresses the overall sentiment in a sentence. The main motivation for this feature is the subjectivity of an ironic utterance, which means it contains a positive or negative opinion. This is represented in the sentiment score feature.

3. Experiment and Results

3.1 Data

The original database[10] used in this paper consists of almost 25 thousand headlines collected from news websites Nu.nl and De Speld. De Speld is a satirical news website, while Nu.nl produces 'real' news. The headlines by De Speld are labeled satirical* (represented with a '1') and the headlines by Nu.nl are labeled non-satirical (represented with a '0'). Furthermore, all headlines are labeled with one of three categories: Political news, domestic news or foreign news. Because some of the headlines are both political and foreign/domestic, they occur more than once in the data. This is not relevant for our research as we do not want some headlines to be more important than others by occurring more than once. After removing duplicates in the data we are left with 13463 news headlines. Table 1 shows some more statistics about the data.

	SATIRE (DE SPELD)	NOT SATIRE (NU.NL)
TOTAL	5103	8359
POLITICAL	982 (19.24%)	2033 (24.32%)
DOMESTIC	2894 (56.71%)	3107 (37.17%)
FOREIGN	1227 (24.04%)	3219 (38.51%)

Table 1: Data statistics

* The original dataset labels this as sarcasm, but De Speld define themselves as a Satirical news website. Therefore, we label their news as satirical.

3.2 Features

The data is pre-processed into a set of features to capture characteristics in satirical and non-satirical headlines. The feature set in this research mostly comes from the work by Hernández Farías et al. (2015)[2] as shown in Section 2. This paper tried to accurately detect irony with these features, but like we described in Section 2, satire and irony are overlapping figurative phenomena with more similarities than differences. With this in mind, we will test how well these features will perform when used to detect satire in Dutch. The rest of this section will further go into these features and how they are implemented.

Simple textual features

The most simple features are statistical features from textual markers. These include the length of a headline (i.e. the number of words in a headline), the number of question marks in a headline, the number of exclamation marks in a headline and the occurrences of numbers in a headline.

Oppositions in text and oppositions in time

These features are equal to the *Counter-factuality* and *Temporal compression* features suggested by Reyes et al. (2013)[13]. Sentences are checked to include terms that hint at opposition or contradiction in text, such as “maar”, “daarentegen” and “echter”. This feature tells us how many of these terms are in a sentence. The temporal opposition feature works the same way. Terms like “vroeger”, “eerder” and “plotseling” indicate a change in time in a sentence. If we come across such a term, we increase the temporal opposition feature of the corresponding sentence. The full lists of terms can be found in the Appendix.

The libraries by spaCy[16] include some NLP applications for Dutch, including a Named Entity Recognizer (NER) and a Part of Speech (POS) tagger.

Named Entity Recognizer (NER)

The Named Entity Recognizer is trained to recognize types of names entities, like a person, organisation or a country. This is used by Burfoot & Baldwin (2009)[3] to calculate their absurdity feature (see Section 2). We did not succeed in copying this implementation, but we did use this application to check if a sentence contains a person or organisation and how many. Our impression is that this is still relevant, as satire usually ridicules a particular person or organisation.

Part Of Speech (POS) tagger

The POS tagger assigns a part of speech to every word in a sentence. This is used to calculate the number of nouns, adjectives, verbs, determiners and numbers in a sentence.

Sentiment score

To calculate this sentiment score, as suggested by Hernández Farías et al. (2015)[2], another database[11] was used that links Dutch words to a valence score. This valence represents a certain positive/negative attitude score, usually referred to as “pleasantness”, that comes with a word. For example, “dood” (death) has a score of 1.38, while “liefde” (love) has a valence of 6.53. Every word in a headline is checked to be in the valences database. If there is a match, we add the corresponding valence score to a list containing all the valences for that sentence. Finally, the mean of the values in the list is calculated to represent the valence of the sentence. Some of the sentences did not include any word from the valence database. These sentence were given a neutral valence, by taking the mean of all valences of all headlines.

To extract all the features from the headline database, we traverse the data twice. The first time, we iterate over the unprocessed headlines. Most features are calculated in this step, and at the same time we create a new list, tokenizing and lemmatizing the headlines and removing capitalization and punctuation. Then, this new list is traversed to create the last features. This last list is needed, for example, for the sentiment score feature. As explained before, the sentiment score is calculated by comparing words in a headline to words in another database. This database consists of Dutch words in standard form and without capitalization. If we come across “Koud” or “koude” we want it to match with “koud” in the other database. This is achieved by creating the second list, with tokenized and lemmatized headlines, also removed from capitalization and punctuation. This list is also used for the contradictions feature, the feature for oppositions in time and the length feature. The feature extraction algorithm takes 6 to 8 minutes to execute.

3.3 Experiment

For the classification of the feature sets as satire or non-satire, we use a Support Vector Machine (SVM) provided by the open source machine learning library Scikit-learn[12]. A Support Vector Machine is a simple supervised machine learning algorithm for classification. It is designed to find a hyperplane that best separates a dataset in two classes. If only two features are used, this hyperplane is a line that linearly divides the feature data. Unseen data is then classified in one of the two classes according to which side of the hyperplane it belongs to. Most classification tasks use more than two features, meaning that the hyperplane is multidimensional and thus more complex than a line. This classifier achieved great results in many related papers, as described in Section 2.

The algorithm is trained on 75% of the data, and then tested on the remaining 25%. This distribution of train and test data gave the best results. Before splitting the data in train and test sets, the data is shuffled randomly to have both sets in proportion with the original dataset. The results for two different feature sets are shown in Table 2. We compare these results to a baseline, which is calculated by taking the average of the results of three dummy classifiers. The first baseline classifier always predicts the most frequent label in the training data, which is non-satire. The second baseline classifier predicts with a probability according to the training set’s class distribution. So, because 62% of the data is non-satire, this classifier predicts non-satire with a probability of 62% and satire with a probability of 38%. The last baseline classifier generates predictions at random. Our main classifier, which uses all features, performs much better than the baseline classifier with an accuracy of 0.758 and an F-score of 0.601. We can also see that the SVM performing on all features is slightly better than the SVM performing on the four features with the best individual performance. The latter trains and tests with the following features: The length feature, question marks feature, nouns feature and determiners feature. The results of the individual tests for every feature can be seen in Figure 1 and Figure 2.

	SVM WITH RBF KERNEL (ALL FEATURES)	SVM WITH RBF KERNEL (FOUR BEST PERFORMING FEATURES)	AVERAGE OF BASELINE CLASSIFIERS (DUMMYCLASSIFIERS)
ACCURACY	0.758	0.731	0.55
RECALL	0.806	0.783	0.25
PRECISION	0.479	0.417	0.30
F-SCORE	0.601	0.544	0.27

Table 2: Results for different Satire Classifiers

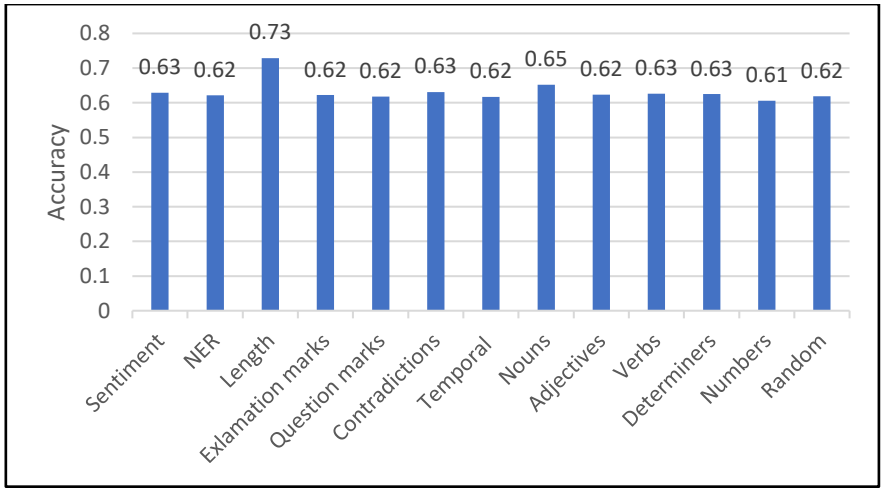

Figure 1: Individual Feature Accuracies

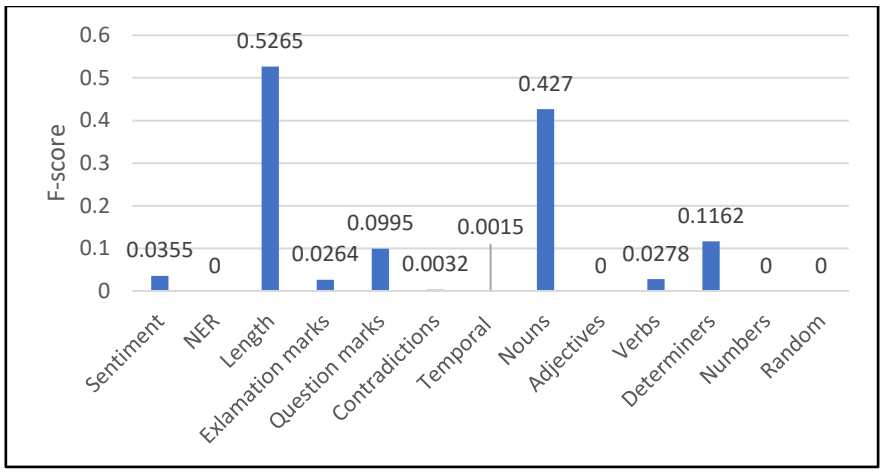

Figure 2: Individual Feature F-scores

For the individual results, the classifier is trained and tested using a single feature. A Support Vector Machine with only one feature looks for a particular value which correctly divides the feature in the two classes. The individual results include a “random” feature, which is zero for every headline. This is to compare the results of the other features to a feature without any relevant information.
The length feature outperforms all the other features with its accuracy and F-score. This means that the length of a headline is the most relevant information to distinguish satire from non-satire in our data. This is verified by looking at the data: The average length of the satirical headlines is 56.1 symbols, while the non-satirical headlines have an average length of 65.8 symbols. The noun feature also stands out with results far above average. Satirical headlines contain an average of 2.064 nouns compared to an average of 2.654 nouns in non-satirical headlines.
Many other features have an F-score of zero, or close to zero. These features do not seem to give any indication for satire or non-satire when individually used. These same features do all have an accuracy of over 61%. This is explained by the statistics of the database. 62% of the database is non-satire, so an accuracy of 62% can be achieved by predicting everything as non-satire. This is exactly what the individual features do: The NER, contradictions, temporal, adjective, number and random features all classify 100% of the data as non-satire. This cannot be seen as an intelligent classifying algorithm. However, these features should not be discarded as useless (except for the random feature). They do contribute positively to the results when used in combination with the other features, because the classifier using all features performs better than the classifier using only the

individually best performing features, as we can see in Table 2. With the length and noun feature, respectively 80% and 79% are classified as non-satire.

	POLITICAL NEWS	DOMESTIC NEWS	FOREIGN NEWS
ACCURACY	0.748	0.718	0.775
F-SCORE	0.463	0.672	0.429

Table 3: Results for using the classifier separately on the different kinds of news

For the last tests, we trained and tested the classifier on the different kind of news separately. The results are presented in Table 3. The database includes news tagged as political, domestic or foreign. The accuracies are around the same percentage, but the classifier achieves a significantly higher F-score on domestic news compared to political of foreign news. This is because of a higher precision, meaning that much more of the data predicted as satire is actually satire (i.e. less false positives).

4. Conclusion and Discussion

Our model is built to detect satire in Dutch news headlines using simple textual features and simple sentiment analysis. It is trained and tested on a corpus of over 13000 news headlines with 38% satire and 62% non-satire. The headlines are represented as vectors of features, with features based on textual markers, grammar, words signalling oppositions in text, words signalling oppositions in time and sentiment. A simple machine learning algorithm tries to distinguish between satire and non-satire using these features. Our algorithm achieved an accuracy of 0.758 and an F-score of 0.608.
In the beginning of the paper we asked ourselves the following question: *To what extent can we detect satire in Dutch news headlines using simple textual features, without the related world-knowledge?* The results of this research are promising for further work on automatic detection of figurative languages in Dutch. As many papers have already shown for English, it shows that a computer can make a reasonable distinction between satire and non-satire in Dutch using simple features. But, the results could be better, especially if we want to use it in high-level sentiment analysis applications. The precision of the classifier is much lower than the recall. This means that a lot of non-satire is predicted to be satire. This is considered to be because of the very subtle satire that closely imitates real news. This leaves very small differences for the classifier to exploit. Another reason could be that real news includes some figurative language as well, or it uses citations with figurative language, which tricks the classifier in predicting it as satire.
In the case of an automated assistive tool that flags misleading or false information, as suggested by Rubin et al. (2016)[17], the low precision of our model is not that bad. This means that some 'safe' data (e.g. real news) would also be flagged by the tool. The user of the tool could double-check this flagged data and see that it is safe. It is much more important to have a high recall, so that as little as possible false information (e.g. fake news) is slipped through the tool as being safe. Thus, our model could work in this situation.
When we look at the individual feature results, the sentiment score feature performed far below the expectations. This feature is designed to capture very positive or very pessimistic/negative cases of satire. Clearly, this does not distinguish it from the real news by Nu.nl. Still, this is only when it is used individually. In Table 2 we do see an increase when using the individually underperforming features together with the better performing features. In hindsight, we think that this feature could be much better represented as Reyes et al. (2013) did with their *Polarity s-grams* (see Section 2s)[13]. This feature captures a switch in polarity, which is more of a unique characteristic of irony and satire than being very positive or very negative.

On the other hand, the length and nouns feature showed very good individual performances. The average length of satirical headlines seemed to be lower than that of real headlines. Furthermore, the satire contained less nouns than the real news, as described in Section 3.3. These could be recurring textual and grammatical characteristics of satirical headlines. Initially, we expected the nouns feature to correlate with the NER feature, as the names of individuals or organisations are represented as nouns in a sentence. However, the nouns feature performs much better than the NER feature. This means that the detected characteristic of satire that the nouns feature represents is purely grammatical. In the last tests, we compared the results of our classifier on the different kinds of news. The classifier showed a much better performance on domestic news than on political and foreign news, with an F-score of 0.672. The only reason for this we could think of is that the satirical news that is political and foreign more often imitates current real political or foreign news. This subtle satire is more often related to world knowledge than domestic news is, which could make the detection of satire more difficult[3] for political and foreign news.

If we compare our results with the results by Liebrecht et al. (2013)[4], which classified Dutch tweets as sarcastic or not, we see that their model performs slightly better than ours. But, the satire by De Speld is much more fine-grained than satire/sarcasm made by Twitter users. De Speld deliberately imitates real news with its satire, which leaves small distinctions between the real and satirical news. Furthermore, these models for detecting sarcasm in Tweets used emoticons or capitalization, for example in ironic sentences like “I HATE to admit it but, I LOVE admitting things”[13], as an indicator for irony. These things are not applicable to detecting the more detailed satire in our corpus. This leaves us with less features to work with.

Some possible problems with the experiment came to mind during our research. The algorithm could be detecting headlines to be from Nu.nl or De Speld, without actually detecting satire. It could be recognizing some pattern in the writing by one of the websites. That would mean that this algorithm could only be used to distinguish between articles from Nu.nl and De Speld. This could also be backed up by the very good performance of the length and nouns feature. We observed that the satirical headlines of De Speld had a significantly lower length than the real news articles by Nu.nl. The classifier uses this to distinguish the satire from the non-satire, but these could just be guidelines for the length of headlines by Nu.nl or De Speld. The same holds for the nouns feature. For future research, different sources should be used to collect satirical and non-satirical news. This way a classifier would be more confident in separating satire from non-satire, instead of just detecting differences between two news sources. Another burden we came across was the available data. For English there is a lot more helpful data available for this kind of research than for Dutch. Reyes et al. (2013) used a lexical database for English for multiple features[13]. Among other things, this database links words to all of its synonyms. It is also used to implement their similarity module which calculates a relatedness score for all pairs of terms in a sentence. This score represents the contextual imbalance of the words in a sentence. With irony, this score is supposed to be higher than with non-irony. Similar tools for the Dutch language are not available, or harder to find, which makes it more difficult to implement some features used by Reyes et al. (2013). A more extensive research could invest time to create databases like these in Dutch. With the right data, features like the profanity and slang features by Burfoot & Baldwin (2009)[3] and the contextual imbalance feature by Reyes et al. (2013)[13] could be implemented for Dutch as well. These features saw great results and we think these would also do very well for Dutch. Another suggestion for future work is to use complete Dutch news articles, like Burfoot & Baldwin (2009)[3], instead of only using the headline of an article. Our impression is that the extra satirical content of the article would make the model more confident to classify the data. Overall, we think that a model could be created with the proposed features in the discussed research that does not need the relevant world knowledge to accurately detect Dutch satire in text.

References

1. Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). *Automatic Sarcasm Detection: A Survey.* Association for Computing Machinery. https://www.cse.iitb.ac.in/~pb/papers/acm-csur17-sarcasm-survey.pdf

2. Hernández Farías, L., Benedí Ruiz, J. M., & Rosso, P. (2015). *Applying Basic Features from Sentiment Analysis for Automatic Irony Detection*. Springer International Publishing. https://riunet.upv.es/bitstream/handle/10251/64255/HernandezFarias-Benedi-Rosso-autor.pdf?sequence=4

3. Burfoot, C., & Baldwin, T. (2009, August). *Automatic Satire Detection: Are You Having a Laugh?* Association for Computational Linguistics. https://www.aclweb.org/anthology/P09-2041.pdf

4. Liebrecht, C., Kunneman, F., & van den Bosch, A. (2013). *The perfect solution for detecting sarcasm in tweets #not*. Association for Computational Linguistics. https://www.aclweb.org/anthology/W13-1605.pdf

5. Weitzel, L., Prati, R. C., & Aguiar, R. F. (2016, March). *The Comprehension of Figurative Language: What Is the Influence of Irony and Sarcasm on NLP Techniques?* Springer International Publishing. https://www.researchgate.net/publication/299470588_The_Comprehension_of_Figurative_Language_What_Is_the_Influence_of_Irony_and_Sarcasm_on_NLP_Techniques

6. Reyes, A., Rosso, P., & Buscaldi, D. (2011, November). *From Humor Recognition to Irony Detection: The Figurative Language of Social Media.* https://riunet.upv.es/bitstream/handle/10251/35314/From%20Humor%20Recognition%20to%20Irony%20Detection.pdf?sequence=8

7. Tepperman, J., Traum, D., & Narayanan, S. (2006, January). *"YEAH RIGHT": SARCASM RECOGNITION FOR SPOKEN DIALOGUE SYSTEMS.* https://www.researchgate.net/publication/221491095_Yeah_right_Sarcasm_recognition_for_spoken_dialogue_systems

8. Tran, P. H., & Huong, P. (2016). *Big Data, Internet Of Thing: new trends in the Digital Marketing era.* https://www.researchgate.net/publication/327449264_Big_Data_Internet_Of_Thing_new_trends_in_the_Digital_Marketing_era

9. Gandomi, A., & Haider, M. (2015, April). *Beyond the hype: Big data concepts, methods, and analytics*. https://www.sciencedirect.com/science/article/pii/S0268401214001066

10. Tuin, H. (2020, December). *Dutch news headlines for sarcasm detection* https://www.kaggle.com/harrotuin/dutch-news-headlines

11. Moors, A., De Houwer, J., Hermans, D., Wanmaker, S., Van Schie, K., Van Harmelen, A., De Schryver, M., De Winne, J., & Brysbaert, M. (2012). *Norms of Valence, Arousal, Dominance, and Age of Acquisition for 4300 Dutch Words.* http://crr.ugent.be/papers/Moors_et_al_BRM_norms_Valence_Arousal_Dominance_AoA.pdf

12. Pedregosa *et al.*, Scikit-learn: Machine Learning in Python JMLR 12, pp. 2825-2830, 2011.

13. Reyes, A., Rosso, P., & Veale, T. (2013). *A multidimensional approach for detecting irony in Twitter*. https://link.springer.com/article/10.1007/s10579-012-9196-x#citeas

14. Reyes, A., & Rosso, P. (2014). *On the Difficulty of Automatically Detecting Irony: Beyond a Simple Case of Negation*. Knowledge and Information Systems. https://riunet.upv.es/bitstream/handle/10251/40330/kaisFinal.pdf

15. Buschmeier, K., Cimiano, P., & Klinger, R. (2014). *An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews*. Association for Computational Linguistics. https://www.aclweb.org/anthology/W14-2608.pdf

16. Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*. https://spacy.io/

17. Rubin, V., Conroy, N., Chen, Y., & Cornwell, S. (2016, June). *Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News.* https://www.researchgate.net/publication/301650504_Fake_News_or_Truth_Using_Satirical_Cues_to_Detect_Potentially_Misleading_Newsv

18. The Media Insight Project. (2014). *The rational and attentive news consumer.* https://www.americanpressinstitute.org/publications/reports/survey-research/rational-attentive-news-consumer/

Appendix

- List of terms hinting at opposition or contradiction in text:
  - maar
  - daarentegen
  - echter
  - integendeel
  - enerzijds
  - anderzijds
  - tegenover
  - hoewel
  - toch
  - ofschoon
  - ondanks
  - anders
  - tenzij
  - desondanks
  - niettemin
  - desalniettemin
  - behalve
  - weliswaar
  - noch

- List of terms hinting at opposition in time:
    - plosteling
    - abrupt
    - nu
    - vroeger
    - later
    - eerder
    - latere
    - eerdere
    - plots
    - ineens
    - onverwacht
    - opeens
    - snel
    - tegelijkertijd
    - nadat
    - daarna
    - wanneer
    - intussen
    - voordat
    - aanvankelijk
    - eerst
    - tijdens

- Link to all the written code and used data for this research:
https://github.com/drprofMatthijs/Simple-Automatic-Satire-Detection-in-Dutch-Headlines