Computational Linguistics in the Netherlands Journal 2 (2012) 34–51 Submitted 09/2012; Published 12/2012

Developing a part-of-speech tagger for Dutch tweets

Tetske Avontuur, Iris Balemans, Laura Elshof, Nanne van Noord, Menno van Zaanen
Tilburg University, Tilburg, The Netherlands

Abstract

In this article we describe the design and creation of a part-of-speech tagger specifically for Dutch data from the popular microblogging service Twitter. Starting from the D-Coi part-of-speech tag set, which is also used in the SoNaR project, we added several Twitter-specific tags to allow the tagging of hashtags, @ mentions, emoticons and URLs. The tagger consists of the Frog tagger combined with a post-processing module that incorporates the new, Twitter-specific tags in the Frog part-of-speech output. Running the Frog tagger and the post-processing module sequentially leads to a part-of-speech tagger for Dutch tweets. Approximately 1 million tweets collected in the context of the SoNaR project were tagged by Frog and the post-processor combined. A sub-set of annotated tweets have been manually checked. Lastly, we evaluated the adapted part-of-speech tagger.

This project was accomplished by eight Master's students from Tilburg University, who had just completed a course in natural language processing. In addition to the theoretical knowledge they acquired during the course, this project, which took approximately a week, offered them hands-on experience.

1. Introduction

Social media sites provide people with an easy and accessible forum to collaborate and share information on the Internet. According to Kaplan and Haenlein (2010) there are six types of social media, namely collaborative projects, blogs and microblogs, content communities, social networking sites, virtual game worlds, and virtual social worlds. In particular, we are interested in microblogs in which users can share information by posting short status updates. As an example, Twitter, a popular microblogging site, allows people to share information in the form of messages with a maximum of 140 characters.

Currently, social media are extremely popular and hence generate huge amounts of data. For instance, Twitter generates approximately 340 million messages (called tweets), per day as of March 2012, which amounts to more than 1 billion tweets every 3 days.[1]

Since social media generate so much user-generated data, it is interesting from a computational linguistic point of view to investigate the potential of extracting useful information from this data. To do this, some enabling technologies are essential. In the field of natural language processing, a large number of tools rely on the availability of morphosyntactic or part-of-speech (POS) information.

In this article, we describe the development of a POS tagger specifically designed for the analysis of Dutch texts found in the user-generated data of the microblogging service Twitter. The data found on Twitter requires special handling, because due to the restriction on the length of the messages, users write succinct messages. This leads, for instance, to users omitting letters or even words, which again may lead to syntactically incorrect sentences. However, typically, reinserting the missing letters or words may be easy to human readers given the context.

The idea of developing a POS tagger for microblogging posts is based on related work by Gimpel et al. (2011), which describes the development of a POS tagger for English tweets. More information about this project can be found in Section 2. Another idea taken from their work is to let a group of students perform the research. More specifically, the group consisted of eight Master's students from Tilburg University who had all just completed a Master's course in natural language processing. The

[1] http://blog.twitter.com/2012/03/twitter-turns-six.html

students have different scientific backgrounds (such as linguistics and computer science) and were local, Dutch, and international students coming from different countries.

The rest of this article is structured as follows. In Section 2, we will provide a brief overview of research that uses POS tags in a microblogging context. Next, in Section 3, we will describe how we created the POS tagger based on an existing system. The POS tagger is evaluated and the results are given in Section 4. The educational experiences (described as a chronological overview of the process) are treated in Section 5 and finally, the conclusions are given in Section 6.

2. Background

As mentioned earlier, the idea for the project described in this article comes from a related project by Gimpel et al. (2011). In this study they address the problem of POS tagging for English data from the microblogging service, Twitter. Starting from scratch, they developed an English Twitter-specific tag set. Using this tag set, they manually corrected English tweets that were annotated using Stanford POS tagger (Toutanova et al. 2003). Additionally, they developed features to build a machine learning classifier that tags unseen tweets. The features relate to Twitter orthography, frequently-capitalized tokens, the traditional tag dictionary, distributional similarity and phonetic normalization. The classifier was trained on the manually annotated data, which consisted of 1,827 tweets. Out of this dataset 1,000 tweets were used for training, 327 for development and 500 for testing purposes.

Several experiments to evaluate the features were conducted. The evaluation was designed to allow the testing of the efficacy of the feature set for POS tagging, focusing specifically on the impact of the performance when limited amounts of training data are available. The tagger with the full feature set led to 89.37% accuracy on the test set.

The project of Gimpel et al. (2011) was accomplished in approximately 200 person-hours spread across 17 people and two months. The aim of the project was to provide richer text analysis of Twitter data and related social media datasets. They also believe that the annotated data can be useful for research into domain adaptation and semi-supervised learning.

Several studies have shown how large amounts of data can be effective. However, even though microblogging services generate large amounts of data, this also includes a large amount of “noise” in contrast to text types such as news articles. According to Sproat et al. (2001) and Ritter et al. (2011) the quality of messages varies significantly. Typos, ad hoc abbreviations, phonetic substitutions, ungrammatical structures and emoticons may have a major impact on the quality of the output of text processing tools.

There are two broad approaches to handling the noisy microblogging data. Firstly, one may try to correct or normalize the text. Secondly, one may try to adapt the natural language processing tools that deal with the noisy data. In this article we will deal with the problem using the second approach. An example of the first approach can be found in Han and Baldwin (2011), which is also based on Twitter data. They propose the task of lexical normalization for short text messages, such as tweets. They describe a method to identify and normalize ill-formed words. Both word similarity as well as context are exploited to select the best candidate for a noisy word token.

Recently, there have been studies that try to perform different types of analyses based on Twitter data. In these studies, POS information is an important information source used in the analysis of different aspects of social networks and Twitter in particular.

In Pak and Paroubek (2010), linguistic information is used for the task of sentiment analysis of English Twitter data. In this work, the researchers show how to automatically collect a corpus for sentiment analysis and opinion mining purposes. They perform linguistic analysis of the collected data in the corpus and explain the discovered phenomena. Using the corpus, they build a sentiment classifier that is able to determine positive, negative and neutral sentiments for pieces of text. One of the types of linguistic information is POS. They observe the difference in distributions of POS tags among data from positive, negative and neutral sets. Results show that certain POS tags might be strong indicators when identifying the emotional content of a text.

Another example of an application of natural language processing in the area of social media, is the study by Tromp (2005). In this thesis, automated sentiment analysis is performed, which extracts opinions from short multilingual texts found in social media. This is done using a four step approach, using POS tagging, language identification, subjectivity detection and polarity detection. The analysis of the results show an overall accuracy of 69.2%. Since sentiment analysis can be used as an automated means to perform marketing research, the results of the automated sentiment analysis are mapped on the results of traditional surveys which are normally used for this purpose. In the end, the study shows that automated sentiment analysis cannot replace traditional surveys.

In both Pak and Paroubek (2010) and Tromp (2005), the authors mention that they use TreeTagger (Schmid 1994) for POS tagging. However, no information is provided about whether the tagger was modified in any respect. It appears that the standard TreeTagger is used and applied to tweets directly using the standard tag set and no additional domain adaptation. It is also unclear whether an adaptation of the POS tagger to tweets will lead to improved results on the task of sentiment analysis.

3. System overview

In this section we describe the POS tagger for Dutch tweets. Firstly, we will describe the tag set that is used to annotate the tweets, followed by a description of the system.

3.1 Tag set

As already mentioned in the introduction and the background section, there are differences between language found in tweets to that found in "regular" text, such as news articles. When analyzing (Dutch) tweets, one can identify specific aspects that distinguish the special nature of tweets as texts in contrast to "regular" text. We summarize those differences as follows:

@ mentions When a user wants to refer to another Twitter user, they use the character `@` before their Twitter user name.

# tags People use the hashtag symbol `#` before relevant keywords in their tweet to categorize those tweets so that they are returned more easily as results of a Twitter search.

Discourse markers Tweets may contain discourse markers like `RT` together with the discourse marker `:` which is used when someone re-tweets another user's tweet. Another example of a discourse marker is a (combination of) symbol(s) that indicate the start of a comment on a re-tweeted tweet, such as | or <<.

Emoticons When users want to express their emotion or feelings in tweets, they can do so using emoticons. An emoticon is a graphical representation, often of a facial expression, consisting mostly of punctuation marks. For instance, `:)` represents a happy face, `:p` represents a face with the tongue stuck out.

URLs When users want to refer to web pages they add URLs to the tweet. This allows, for instances, addition of multimedia, such as images, to tweets by linking to on-line web pages containing such multimedia data.

Alternative grammar and spelling Probably due to the limited length of a tweet (of at most 140 characters), tweets often lack coherence. Also, they are sometimes written with limited grammar and non-perfect spelling.

Table 1: Tag set as used in Gimpel et al. (2011).

	Nominal, Nominal+Verbal		Twitter/online-specific
N	Common noun	#	Hashtag
O	Pronoun	@	@ mention
S	Nominal+possessive	˜	Discourse marker
ˆ	Proper noun	*U*	URL or email address
Z	Proper noun+possessive	*E*	Emoticon
L	Nominal+verbal		
M	Proper noun+verbal		
	Other open-class words		**Miscellaneous**
V	Verb including copula, auxiliaries	$	Numeral
A	Adjectives	,	Punctuation
R	Adverb	*G*	Other abbreviations, foreign words possessive endings, symbols, garbage
!	Interjection		
	Other closed-class words		
D	Determiner		
P	Pre-or postposition or subordinating conjunction		
&	Coordinating conjunction		
T	Verb particle		
X	Existential there, predeterminers		
Y	*X*+verbal		

As an example of these differences, consider the tweet `Het ligt echt niet aan mijn rijstijl dus | RT @dgnijmegen: Auto van burgemeester rijdt paal eruit: #NIJMEGEN - http://bit.ly/ffj9TN`. Here, the user provides some comments, found at the beginning of the tweet in reply to another tweet. The boundary between the comments and the original tweet is marked by the discourse marker: `|`. The second part of this tweet consists of a re-tweet, marked by the discourse marker: `RT`, followed by an @ mention of the author of the original tweet, followed by another discourse marker: the colon. Next, the original tweet can be found, which contains the hashtag `#NIJMEGEN`.

Since tweets may contain these specialized constructions that do not occur in most other text types, we need to select or design a Twitter-specific tag set. We started by analyzing the tag set proposed by Gimpel et al. (2011). This tag set, which can be found in Table 1, is specifically designed for English tweets and contains tags that are language specific. It turned out that these tags describe situations that often occur in English Twitter data. One example is the *L* tag which describes the situation where nominal and verbal tokens are glued together into one token. Examples of such tokens in English are: `I'll`, `work's`, or `you're`. Analyzing the Dutch tweets that are used in this research (which will be described in more detail in Section 4.1), showed that this construction does not occur in the Dutch data used in this task.[2]

Removing the tags from the English tag set that are irrelevant to Dutch leaves only a very small number of useful tags. Because of this, we decided to reuse the Twitter-specific tags and incorporate these in an existing tag set for Dutch.

Recently, several corpus collection and annotation projects that concentrate on Dutch language data have finished (or are nearly finished) and it turns out that all of them build on the same POS tag set. The tag set we use here is, for instance, also used in the SoNaR corpus project (Oostdijk et al. 2008, Oostdijk et al. In press).

[2] However, in the analysis of the dataset that we performed afterwards as described in Section 4.3.2 indicates that the dataset we used may be significantly different from typical tweets, where constructions as `kheb` (`I+have`) do occur.

SoNaR stands for Stevin NEderlandstalig Referentiecorpus. The aim of this project is to build a large collection of written contemporary Dutch (and Flemish) texts. SoNaR is a collaboration between Radboud University Nijmegen, Tilburg University, University of Twente, Utrecht University, and KU Leuven. It is funded by the Dutch-Flemish Stevin[3] program. The annotation of morphosyntactic information is based on the tag set developed by the D-Coi (Dutch Language Corpus Initiative) project (Oostdijk and Boves 2006, Oostdijk 2011).

The D-Coi tag set consists of 320 individual tags which are grouped into 12 main tag groups. Each tag consists of a main tag and can be made more specific by adding arguments. Take, for example, the tag *N(soort,mv,dim)*. Here, *N* indicates that the word that is annotated with this tag is a noun. The arguments *(soort,mv,dim)* specify that the noun is a generic noun (*soort*), in plural form (*mv*) and is in the diminutive shape (*dim*). More detailed information on the extensive D-Coi tag set can be found in Van Eynde (2005).

The POS tagger described in this article analyzes tweets and outputs a tag sequence with tags from the full D-Coi tag set, which is extended with the Twitter-specific tags that can be found under the heading Twitter/online-specific in Table 1, which are renamed as *HASH*, *AT*, *DISC*, *URL*, and *EMO* respectively.

3.2 System description

A POS tagger reads in a sequence of words to be POS tagged and for each token in the sequence assigns a POS tag from the tag set. Because we have created a new tag set (which is based on a detailed existing tag set), we need to either develop a completely new tagger or modify an existing one to handle the new Twitter-specific tags. First, we will describe the POS tagger we have used and then describe a post-processing module that modifies the output of the POS tagger in order to incorporate the Twitter-specific tags.

3.2.1 Frog

The POS tagger that is used in the SoNaR project is called Frog (formerly known as Tadpole). This is a POS tagger that has been trained to produce output containing tags from the full D-Coi tag set. A preliminary, informal, experiment using Frog indicated that the output of the program is of such quality that we expected that it could be used successfully to POS tag Tweets.

Even though we have a pre-trained POS tagger (Frog) available, we still need to make sure that the Twitter-specific tags are also taken into account as possible output. Essentially there are two approaches to incorporating the new tags. We can either retrain the tagger using data that is manually annotated using the new tag set or by by implementing an add-on phase, which modifies the output of Frog. Since only a small number of tags are added to the D-Coi tag set and that for most of these a rule-based mapping can be created, we decided to approach the addition of the Twitter-specific tags using a post-processing module that takes the output of the Frog tagger as input.

The modified tagger works as follows. Firstly, we tag the tweets with the Frog POS tagger. Frog first tokenizes the tweets using UCTO[4]. Next, the memory-based tagger, MBT, is applied to the tokenized sequence, which assigns POS tags. Frog can also perform additional processing, such as morphological analysis, named-entity recognition and syntactic parsing.

The output of Frog is a list of lines, each consisting of nine tab-delimited columns. Each line denotes information of one token. An example of the output can be found in Table 2. The columns contain the following information:

1. A incremental token number, which resets each sentence;

[3] `http://lands.let.ru.nl/projects/SoNaR/`

[4] `http://ilk.uvt.nl/ucto/` Note that the version of UCTO used at the time of the experiments could not properly handle urls, hashtags and @ mentions. However, in more recent versions of UCTO can handle urls, hashtags and @ mentions properly if the twitter configuration is selected explicitly.

Table 2: Output of the sentence `Marie vroeg zich af of hij nog zou komen.` generated by Frog in the column format. We only use the second column (tokens) and the fifth column (POS tag).

1	`Marie`	Marie	[Marie]	SPEC(deeleigen)	1.000000	B-NP	2	su
2	`vroeg`	vragen	[vraag]	WW(pv,verl,ev)	0.532544	B-VP	0	ROOT
3	`zich`	zich	[zich]	VNW(refl,pron,obl,red,3,getal)	0.999740	B-NP	2	se
4	`af`	af	[af]	VZ(fin)	0.996853	O	2	svp
5	`of`	of	[of]	VG(onder)	0.733333	B-SBAR	2	vc
6	`hij`	hij	[hij]	VNW(pers,pron,nomin,vol,3,ev,masc)	0.999659	B-NP	8	su
7	`nog`	nog	[nog]	BW()	0.999930	B-ADVP	8	None
8	`zou`	zullen	[zal]	WW(pv,verl,ev)	0.999947	B-VP	5	body
9	`komen`	komen	[kom][en]	WW(inf,vrij,zonder)	0.861549	I-VP	8	vc
10	`.`	.	[.]	LET()	0.999956	O	9	punct

2. The token itself;
3. The lemma of the token, which is based on the output of the memory-based lemmatizer MBLEM[5]);
4. The morphological segmentation, which is also based on the output of the memory-based morphological analyzer MBMA[6]);
5. The POS tag corresponding to the token, which may contain tags from D-Coi tag set and is assigned by the memory-based tagger MBT[7]) (Daelemans et al. 1996b);
6. The confidence of the system in the choice of the tag, which is a number between 0 and 1, representing the probability mass reserved for the tag with the highest probability in the total distribution over all tags according to the classifier;
7. The output of the chunker or shallow parser on the basis of MBT;
8. The token number identifying the head word of the dependency of the current token, which is provided by the constraint-satisfaction inference-based dependency parser, CSI-DP;
9. The type of dependency relation of the current token with its head word.

3.2.2 Retokenization and mapping

From the output provided by Frog, we are only interested in the token (column 2) and POS tag (column 5). Based on this information, a post-processing module changes one or more POS tags output by Frog into one of the Twitter-specific tags (*HASH*, *AT*, *DISC*, *URL*, and *EMO*). The possible transformations into the Twitter-specific tags are described using a set of regular expressions.

In most cases, the output of Frog needs to be retokenized when assigning the Twitter-specific POS tags (in fact, all retokenized tokens receive a Twitter-specific tag). UCTO, which provides the tokenized input of Frog, separates the stream of characters into tokens. However, punctuation serves as an indicator for a new token. This means that @ mentions and hashtags are broken into the indicator part (`@` and `#` respectively) and the remainder of the @ mention and hashtag. Additionally, URLs are broken into parts (if they contain `http://` for instance, but also if they

[5] `http://ilk.uvt.nl/mbma/`
[6] `http://ilk.uvt.nl/mbma/`
[7] `http://ilk.uvt.nl/mbt/`

Table 3: Mapping from output of Frog in the column format to Twitter-specific output in the column format. The output in the Twitter column has been aligned to the entries in the left column for clarity reasons.

Frog		Twitter	
RT	SPEC(symb)	RT	DISC
@	SPEC(symb)		
nilicule	ADJ(prenom,basis,met-e,stan)	@nilicule	AT
:	SPEC(symb)	:	DISC
#	SPEC(symb)		
sdgeld	WW(vd,vrij,zonder)	#sdgeld	HASH
http://t.co/74h22oo	SPEC(deeleigen)	http://t.co/74h22oo	URL
:	LET()		
-	LET()		
)	LET()		
)	LET()		
)	LET()	:-)))	EMO

contain other characters, such as `=`), just as emoticons (which contain mostly of punctuation) and discourse markers.

Retokenizing @ mentions, hashtags, URLs, and discourse markers is relatively straightforward. In the original input, all tokens separated by UCTO should be glued together until a space is found. The retokenization requires processing of the stream of tokens in parallel to the stream of characters. For instance, if `C# programming` is a part of a tweet, which is tokenized as `C␣#␣programming`, the token `#` belongs to the token `C` and does not indicate a hashtag. If we would not process the stream of characters in parallel, the program would change this into `#programming` and assign a *HASH* tag to it. URLs are only retokenized if they start with the `http://` marker. This is done to make sure that non-URL text containing a period without trailing white space is not considered a URL.

However, to be able to identify emoticons (in contrast to other combinations of punctuation), we manually created a list of 156 emoticons that we encountered in a selection of tweets. This list also includes emoticons that can vary in length such as `;-))))))))`, which are represented using regular expressions, such as `;-)+` indicating that the closing bracket may be present one or more times. Other emoticons cannot easily be generalized like this, such as `\o/`. Furthermore, the reverse of the emoticons (for instance `(:` versus `:)`) are added since some users tend to write emoticons the other way around.

After retokenization, the post-processing module identifies the tokens that require Twitter-specific tags. The *AT* tag is assigned to tokens starting with the `@` symbol followed by additional characters. Similarly *HASH* is assigned when a token starts with the `#` symbol and is followed by more characters. The *URL* tag is assigned if the token resembles a URL and a token receives the *EMO* tag the token matches any of the emoticon regular expressions.

Table 3 shows an example of how the Frog output is converted into the Twitter-specific format. The empty lines in the right-hand column are introduced to align the Twitter-specific tokens to that in the Frog output column. These newlines are not present in the actual output.

4. Experiments

To investigate the quality of the output of the Twitter POS tagger, we conducted several experiments. We first manually corrected the output of the Twitter POS tagger to produce a Gold standard. Next, we compared the output of the POS tagger to this Gold standard. We will describe properties of the dataset, the annotation process and the actual quality of the output of the Twitter POS tagger.

4.1 Dataset

Tweets are easily available, as they can be downloaded directly (in XML or JSON format) from the Twitter site. However, evaluating the POS tagger on dynamically downloaded tweets makes an exact replication of the evaluation hard if not impossible. For this reason, we prefer to use a "standard" tweet collection. Fortunately, within the context of the SoNaR project (already mentioned in Section 3.1) 1,074,360 Dutch tweets had already been collected. Manual analysis indicated that only a small sub-set of these are non-Dutch. The large majority of the tweets is in Dutch.

The tweets from the SoNaR project come in a format that includes some meta data, such as time stamp, re-tweet information and any URLs that can be found in the tweet. For POS tagging purposes, the only information we extracted is the actual text of each tweet. This allows us to discard all other information (but it is trivial to link the additional information with the POS tagged version of the tweets).

An example of a Dutch tweet from the SoNaR tweet dataset is

```
@marineltwit Dat is dan jammer, toch ??  RT:sommige mensen hebben geen gevoel
voor humor #fe11 :s
```

which translates to

```
@marineltwit That is a pity, right ??  RT:some people do not have a sense of
humor #fe11 :s
```

This tweet contains @ mentions (`@marineltwit`), a discourse marker (`RT:`), a hashtag (`#fe11`), and an emoticon (`:s`).

4.2 Annotation

To be able to evaluate the output of the POS tagger, a collection of correctly annotated tweets is required. This allows for a comparison of the output of the POS tagger against this Gold standard. Since we have created a POS tag set specifically designed for Dutch tweets, no such Gold standard corpus exists. This means that we need to manually build a correctly annotated Gold standard POS tagged Twitter corpus. The Gold standard is created by three native Dutch speaking annotators. The annotators are communication and information science students who have affinity with linguistics. Based on the D-Coi annotation guidelines (Van Eynde 2005), extended with guidelines for the annotation of the Twitter-specific tags, all three annotators each manually corrected the generated POS output.

The annotators manually checked the output of the POS tagger and modified any tag they considered incorrect. If all annotators agreed on the same tag, this tag was considered correct (or corrected). The cases in which the annotators did not agree, they discussed the situation and in practice this came down to a majority vote between human annotators leading to the ultimate Gold standard tag. In the cases where all annotators initially disagreed, a consensus was reached.

In total 1,056 tweets consisting of 16,881 tokens were taken from the SoNaR dataset for manual correction. This subset was created by selecting Twitter users that have at most 100 tweets in the dataset. This comes down to tweets from 23 different users who provided on average 46 tweets each.

To make sure the annotators did not accidentally modify the text of the tweets and only selected tags that are valid within the tag set, we decided to use annotation software for the correction of the POS tagger output. We decided to use the annotation tool from Gate[8] (Cunningham et al. 2011) for this purpose. This tool allowed the annotators to have a visual overview of the valid tags in the tag set and the output of the process was guaranteed to be a file in a valid format.

[8] http://gate.ac.uk/

Table 4: Inter-annotator agreement results of the Gold standard POS tags between annotators in pairs (Cohen's Kappa) and all annotators combined (Fleiss' Kappa). The results in the *all* part describe inter-annotator agreement results of all manually annotated tokens. The results in the *modified* part describe the inter-annotator agreement results of those tokens that were modified by at least one annotator.

Tokens	Measure	Annotators	Score
All	Cohen's Kappa	A vs. B	91.20
		A vs. C	91.65
		B vs. C	93.32
	Average Cohen's Kappa		92.06
	Fleiss' Kappa		92.06
Modified	Cohen's Kappa	A vs. B	26.13
		A vs. C	27.19
		B vs. C	38.13
	Average Cohen's Kappa		30.48
	Fleiss' Kappa		30.06

4.3 Quantitative results

Before providing the performance of the final POS tagger on the Gold standard, we consider the quality of the annotation of the Gold standard. Additionally, we will investigate whether the language in the Twitter dataset we have used is indeed very dissimilar to other genres of language. Finally, we will provide the results of the evaluation the POS tagger.

4.3.1 Annotator quality

Manually correcting the POS tags for tweets may be a difficult task. There are 325 distinct tags (320 from the D-Coi tag set and 5 Twitter-specific tags) to consider for each token. Computing inter-annotator agreement provides some insight in the complexity of the task. If it is the case that all three annotators always agree, the task can be considered clearly defined and perhaps (relatively) easy. However, if the three annotators disagree often, the task is not defined clearly or is complex.

We show two different scores that measure inter-annotator agreement; the Cohen's Kappa and the Fleiss' Kappa. The Cohen's Kappa compares the tags assigned by two annotators at a time. Since we have three annotators, this leads to three results. The pair-wise inter-annotator results can be found in Table 4 together with the average Cohen's Kappa inter-annotator agreement score over the three pairs. The Fleiss' Kappa can be used to compare more than two annotators at the same time. We used it to compute the overall inter-annotator agreement, which can also be found in Table 4.

The annotators investigated the output of the POS tagger and modified incorrect tags. This task is different from fully manually annotation of the data from scratch. In Table 4 we first present the inter-annotator agreement results based on all annotated (or manually corrected) tokens. Note that this measure incorporates results from the system (although the annotators were allowed to modify each tag if they thought that was necessary). The second part of the table shows inter-annotator agreement results when considering only the tokens that were modified by at least one annotator. This amounts to 1,981 tokens out of the 16,881 tokens that have been annotated in total, which lead to a different POS tag for 871 tokens in the Gold standard compared to the Frog output. Clearly the results in the second part of the table are much lower as these are the tokens where the annotators disagree with the system. These may be exactly the tokens for which it may be more difficult to identify the right tag (for the system as well as for the annotators).

Even though there may be some discussion on exactly how to interpret these values, the overall inter-annotator agreement scores show consistently high values, which leads us to conclude that there was high agreement amongst the annotators for many "easy" tokens. These inter-annotator agreement results are comparable to those on the English task (where the inter-annotator agreement was 92.2% compared to 92.06% for Dutch) where also the output of a tagger was corrected. However, the more difficult tokens (for which at least one of the annotators disagreed with the output of the system), the inter-annotator agreement is much lower. This indicates that the annotation of these tokens is harder.

4.3.2 Dataset quality

Before we analyze the performance of the POS tagger, we investigate some properties of the dataset, which could give us some idea about what performance we may expect of the system. If tweets are very similar to "regular" text, we may expect high quality output because Frog has been trained on non-Twitter data, but if the tweet data is very different, performance may be significantly lower.

To evaluate the quality of the language used in the Twitter dataset, we identify the words from the tweets that can also be found in the SoNaR corpus. We have used the final SoNaR corpus before curation[9]. This corpus consists of over 500 million tokens which comes down to over 5 million types. The entire corpus was used apart from the section containing tweets. This gives us an indication on the choice of words in Dutch tweets compared to the variety of words in non-Twitter text.

Based on the tokenized tweets of our dataset, we compare both the types (unique words) and tokens found in the tweets against the words in the full SoNaR corpus in a case-insensitive manner. Out of the 483,546 types that can be found in the tweets, 107,046 can also be found in SoNaR. This means that 77.86% of the types that occur in the Twitter dataset cannot be found in SoNaR. Performing the same comparison on the basis of tokens, we see that 7,345,850 tokens out of the 12,513,112 tokens can be identified in SoNaR. This means that on the basis of tokens, 41.29% of the tokens occurring in the tweets cannot be found in the SoNaR corpus. This indicates that the language found in tweets is quite different from the language found in other text types.

At the end of the project, we evaluated whether the SoNaR tweets are a representative sample from Dutch tweets, we conduct a similar experiment, which performs the same evaluation, but now based on randomly gathered collection of Dutch tweets. We created five randomly selected datasets of one million Dutch tweets. On average the datasets contain 139,191 types and 999,965 tokens.[10] Out of these, 21,306 types and 694,160 tokens can be found in the SoNaR corpus. This means that in these dataset 84.68% of the types and 30.58% of the tokens cannot be found. The amount of words in the randomly selected datasets is significantly different ($p < .001$) from the words in the SoNaR tweet dataset. This holds for both the amount of types and tokens as well as the percentage of types and tokens that can be found in the full SoNaR dataset (without the tweets). In other words, the SoNaR tweet dataset, which is for a large part created from tweets by well-known people is significantly different from random tweets. This means that the accuracy scores given in Table 5 are likely to be an overestimation.

4.3.3 POS tagger quality

The performance of the Twitter POS tagger is computed by comparing the output of the tagger against the manually corrected Gold standard. A total of four evaluations were conducted, the results of which can be found in Table 5. Firstly, an evaluation is performed on the entire Gold standard dataset with detailed POS tags (in other words, the full D-Coi tag set extended with the Twitter-specific tags). Secondly, the same evaluation is performed, but this time, the detailed tags are mapped onto their simplified tag. This means that for each of the complex POS tags, such as

[9] Access to this corpus was provided by Martin Reynaert. We are very grateful for his help.

[10] It turns out that the collection of tweets also contain non-Dutch tweets. Due to the characters used in the non-Dutch tweets, the entire tweets are often seen as one token.

Table 5: Accuracy results of the modified Frog POS tagger for Twitter data. Results are given for all tokens in the Gold standard dataset (complete) and for only the modified tokens (modified) as well as for the detailed tag set and the simplified tag set.

		Accuracy
Complete	detailed tags	94.84
Complete	simplified tags	95.58
Modified tokens	detailed tags	61.18
Modified tokens	simplified tags	61.40

N(soort,ev,basis,zijd,stan), only the main POS tag is used. In this case, the tag would be *N*. For the third and fourth evaluation only the tokens that are tagged differently by *at least one* of the annotators are taken into consideration. In Table 5 these results are referred to as modified tags. The sub-set of the tokens that have been changed by the annotators is smaller than the full dataset: 1,981 out of a total of 16,881 tokens in the full Gold standard dataset. Similarly to the first and second evaluation, the third evaluation makes use of the detailed tags while the fourth uses the simplified tags.

The results show that overall the POS tagger (Frog output converted into a Twitter-specific tag set) performs very well. However, if we only consider the tokens for which at least one of the human annotator indicated the token is incorrect, the performance is lower. This is to be expected as these might either be incorrectly tagged tokens, or tokens of which the POS tag is unclear (for at least one of the annotators).

As a further analysis, we provide the confusion matrix in Table 6. This compares the POS tags from the tagger (found on the columns) and the Gold standard tags (found on the rows). Only the course-grained tags are provided due to space constraints. As can be seen, the tagger works well for most tags and hardly any structural mistakes are made. Only on the *N* and *SPEC* tags are confused more regularly. We will take a closer look at this in Section 4.4.

When we compare these results against other POS tagging systems, we find, for instance, in Daelemans et al. (1996a) (which describes the MBT tagger that is also present in Frog) an accuracy of 97.1% for known words, 71.6% for unknown words and an overall accuracy (of known and unknown words combined) of 95.7% based on data from the written part of the Eindhoven corpus (Uit den Boogaart). This corpus has been manually annotated using the WOTAN tag set consisting of around 250 tags (Berghmans 1994). Daelemans et al. (1996a) also mentions that the WOTAN tagger on the same corpus leads to an accuracy between 89.5% and 91.5% on newspaper articles. Note that this accuracy is found when evaluating on a simplified tag set (using only the main tags). Van den Bosch et al. (2007) apply the MBT tagger to Spoken Dutch Corpus and find accuracies of 96.5% using the full tag set and 98.6% using only the main tags.

4.4 Qualitative results

In addition to the quantitative results of the POS tagger and the datasets used, we also identified qualitative properties of the dataset. Furthermore, during the annotation process, certain problems arose, which we will describe in this section.

During manual annotation, three more regularly occurring problems within the system were encountered by the annotators. Firstly, tokenization problems often made it hard to identify the full URLs. UCTO tokenized parts of URLs (which typically include punctuation), which led to whitespace within elements of URLs. In some cases, URLs are not easy to recognize, which means that our retokenization module could not correct this. An example of such a URL is `echtbroodjeaap␣.␣nl`. Since in this case no `http://` is present, the retokenization module does not recognize this as a

Table 6: Confusion matrix between tags from the POS tagger (columns) and the tags of the Gold standard (rows).

	LET	*SPEC*	*N*	*VNW*	*WW*	*AT*	*LID*	*VZ*	*HASH*	*ADJ*	*URL*	*VG*	*BW*	*TW*	*DISC*	*TSW*	*EMO*
LET	3069	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SPEC	3	1214	89	15	23	0	2	6	0	13	0	3	4	14	0	0	0
N	0	352	2318	0	10	0	0	0	0	14	0	0	6	0	0	0	0
VNW	0	14	4	1170	0	0	0	0	0	0	0	0	0	0	0	0	0
WW	0	10	36	0	1798	0	0	0	0	2	0	0	1	0	0	0	0
AT	0	0	0	0	0	420	0	0	0	0	0	0	0	0	0	0	0
LID	0	1	1	1	0	0	868	0	0	0	0	0	4	0	0	0	0
VZ	0	3	5	0	0	0	0	1654	0	0	0	1	0	0	0	0	0
HASH	0	3	0	0	1	0	0	0	547	0	0	0	0	0	0	0	0
ADJ	0	6	9	0	2	0	0	0	0	1049	0	0	0	0	0	0	0
URL	1	1	7	1	0	0	0	0	0	1	295	0	0	0	0	0	0
VG	0	0	0	0	0	0	0	0	0	0	0	422	0	0	0	0	0
BW	0	4	8	0	1	0	1	1	0	10	0	0	927	0	0	0	0
TW	0	7	0	0	0	0	0	0	0	0	0	0	0	236	0	0	0
DISC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	67	0	0
TSW	0	19	21	0	5	0	0	0	0	7	0	1	4	0	0	39	0
EMO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25

URL. Due to this tokenization problem, Frog annotated the different parts separately instead of as part of the URL.

Secondly, a difficulty was found with the tag that was used for names (*SPEC(deeleigen)*). Although in many cases this tag was assigned to tokens correctly, sometimes this tag was too generic and a more specific tag should have been selected. In particular, proper nouns (*N(eigen, . . .)* would have been more appropriate. This tag can be further specified, providing more information about the token such as gender, number, etc.

Finally, the system sometimes failed to recognize emoticons correctly. In some cases emoticons were not recognized where they should be recognized (false negatives). This was probably due to the fact that emoticons are used very creatively in tweets, which implies that a rather long list (of regular expressions) of emoticons (or a completely different system that identifies emoticons) is required in the system. In other cases, the system identified an emoticon which is not a true emoticon (false positive). For instance, emoticons are found in places that do not practically allow for emoticons, such as within a URL (`http://...=)`).

5. Educational aspects

The idea of investigating the possibility of designing and implementing a POS tagger for Dutch tweets stems from the work by Gimpel et al. (2011). The intention in their work is to develop a POS tagger for English tweets in a limited amount of time (a week) collaboratively using a small group of people. The research described in this article is performed in a similar way.

This section describes aspects of the project that are of educational nature. First, we will describe the setup of the project, followed by a section that describes the choices made by the students during the project. Finally, we discuss the experiences of the students.

5.1 Project setup

At the Tilburg University School of Humanities a Master's level course called Natural Language Processing is taught. During this course, the students study theory and basic components used to approach various aspects of natural language processing problems. The lectures have a theoretic part, in which the underlying theories are described. Additionally, during the lecture, students try to solve several exercises, which are discussed afterwards.

Even though the course contains a number of assignments and exercises, students do not experience the problems occurring in a more natural setting that requires natural language processing techniques. To provide the students with a more realistic setting, a problem (that could be tackled in approximately a week) was devised for the students to tackle.

Participation in this project was voluntary. Apart from providing the students with background in the area of natural language processing, there is no further relationship with the course. To allow students to allocate a specific amount of time for the project beforehand, the project was restricted in time. We decided to allocate one week in person-hours for the Twitter POS tagger problem. This, again, was based on the experience from Gimpel et al. (2011).

The reason why the project should last approximately a week is that this should give enough time to perform research or build a working system if enough people participate, but at the same time, the relatively short duration of the project means that the project does not clash with the work students have to do for their program. As such, this project is additional work, in addition to the hours the students spend on their studies.

All students participating in the project have followed the natural language processing course. This gave them a solid background in the material, even if their academic or cultural background varied. In particular, the group consisted of national (Dutch) and international students with computer science, or linguistics backgrounds.

In the end, eight students actively participated in the project. Each was assigned to particular sub-tasks which corresponded to their academic background. Some students were involved in multiple sub-tasks. The different sub-tasks and the different choices made within these sub-tasks will be described in the next section.

5.2 Choices

The project was divided into several sub-tasks. Firstly, a suitable tag set had to be selected or designed. Secondly, a (trainable) POS tagger had to be found. Thirdly, the Gold standard dataset had to be annotated. Some additional tasks are also required, such as the implementation of conversion, analysis, and evaluation scripts. These additional tasks were rather straightforward and will not be described in more detail here.

Note that the sub-tasks have been performed independently in small groups. This allowed the students to perform some of these tasks in parallel (in particular, the development of the tag set and the selection of the POS tagger). In the end, the students found that these sub-tasks depend on each other greatly as the choice of the tag set has an impact on the choice of the POS tagger and vice versa.

The first sub-task the students had to tackle was the selection or creation of a POS tag set. As a starting point, the tag set of Gimpel et al. (2011) was used. Dutch tweets from the SoNaR project (which were already available as they were used in another project), the choice was made to keep the Twitter-specific tags and select an existing Dutch POS tag set. This led to the choice for the D-Coi tag set (Van Eynde 2005).

Given the time restriction of the project and the expected difficulty of the task given the extensive tag set, it was decided that a coarse-grained version of the tag set was to be used. Effectively, a sub-set of the D-Coi tag set was created that still provides useful information about the POS of the text, for example for sentiment analysis purposes, and at the same time was expected to be easier to use while annotating. Another potential advantage of reducing the size of the tag set is that an

Table 7: The initial proposal of the POS tag set which consists of the main tags from the D-Coi tag set, expanded with more specific *N* and *SPEC* tags (in the left column). The right column contains the Twitter-specific tags.

Generic		Twitter	
ADJ	Adjective	*AT*	@ mention
BW	Adverb	*DISC*	Discourse marker
LET	Punctuation	*EMO*	Emoticon
LID	Determiner	*HASH*	# tag
N(eigen)	Proper noun	*URL*	URL
N(soort)	Common noun		
SPEC(afgebr)	Partial words		
SPEC(onverst)	Incomprehensible words		
SPEC(vreemd)	Foreign words		
SPEC(deeleigen)	Part-of-whole words		
SPEC(afk)	Abbreviations		
SPEC(symb)	Symbols		
TSW	Interjection		
TW	Cardinal/ordinal		
VG	Conjunction		
VNW	Pronoun		
VZ	Preposition		
WW	Verb		

increase in the overall system performance was expected as there are fewer classes (tags) to classify into.

Given these considerations, the full D-Coi tag set was reduced by selecting only the main tags. Doing this leads to a tag set of 12 tags. In this tag set, some aspects that may be highly relevant in situations, such as sentiment analysis, are not identified. For instance, the distinction between common and proper nouns cannot be made as only the generic tag for nouns (*N*) is available. To resolve this problem the generic *N* tag (nouns) were replaced with the sub-tags: *N(eigen)* for proper nouns and *N(soort)* for common nouns. Additionally, the *SPEC* tag (for special tokens) was expanded into seven sub-types found in the D-Coi tag set, which allow for the distinction between symbols, abbreviations, incomprehensible words, etc. (The D-Coi tag set also recognizes the *SPEC(meta)* tag, for meta data, but this is not relevant to Twitter data and as such is not added.) The resulting tag set (the sub-set taken from D-Coi together with the Twitter-specific tags) can be found in Table 7.

At the same time, the students searching for a suitable POS tagger identified Frog as a potentially useful POS tagger for this project. Frog assigns POS tags from the full D-Coi tag set. This information was shared with the students working on the development of the POS tag set for the project. In the end, the choice was made to use the full D-Coi tag set combined with the Twitter-specific tags as described in Section 3.1.

Once the tag set and the POS tagger were selected. Students started developing the Gold standard dataset. To ensure that the dataset would not contain any invalid POS tags, it was decided to use annotation software. At first, the Callisto[11] (Day et al. 2004) annotation tool was considered. After loading the tag set, it turned out that the graphical user interface of Callisto cannot handle large amounts of tags. The program shows a list of tags that can be selected for annotation, but the list does not fit on the screen. This makes it unsuitable for our purposes.

[11] `http://callisto.mitre.org/`

Next, MMAX2[12] (Müller and Strube 2006) was tried. Even though MMAX2 can handle large tag sets, it turned out to be difficult to use in practice as changing a tag for a token required several operations.

Finally, the annotation tool provided by Gate[13] (Cunningham et al. 2011) was used for the correction of the annotated tweets. The only disadvantage we found is that Gate allows for the changing of the actual text of the tweet. Even though the annotators were instructed not to modify the text, this sometimes happened by accident without the annotators noticing it, leading to inconsistent data.[14]

5.3 Student experience

During the project, students were sometimes surprised that not everything works as intended (which is what one might expect from the discussed techniques during the course). For instance, the selection of a tool for the manual annotation turned out to be more difficult than expected. Several tools have practical limitations that made them unsuitable for this particular task and textbooks do not always discuss these practical problems.

At the end of the research, the students presented their work at the CLIN conference. Since they enjoyed the project and the presentation so much, they also requested to present their work again when an international researcher visited who wanted to know what our students were doing. A preliminary version of the research presented here was also sent to an LREC workshop on user-generated content (Aminian et al. 2012). Overall, this project changed their view (in a positive way) on how academic research is done. As additional impact of this project, several of these students are now interested in pursuing a PhD.

Finally, students who decided to follow the natural language processing course the year after heard about this project and were requesting that again such a project be organized. Just like the first time, several students were interested.

6. Conclusions

The data available through social media has grown worldwide. In the Netherlands, in particular, social media are extremely popular. For instance, in 2011 the Netherlands had the highest Twitter penetration worldwide.[15] From this popularity, it follows that huge amounts of short Dutch texts (tweets) written by users are publicly available. The SoNaR Dutch corpus building project took advantage of this by incorporating a sub-corpus of Dutch Tweets in the main corpus.

One of the enabling technologies in the area of natural language processing is a POS tagger. This assigns morphosyntactic information to tokens in texts. Since tweets have somewhat different properties compared to, for instance, news articles, a specialized POS tagger for Dutch tweets is required.

In this article, we present a POS tagger for Dutch tweets. The tag set used is an extended version the tag set originally developed within the D-Coi project. The extension allows for the tagging of Twitter-specific tokens. The implementation of the POS tagger is based on Frog, a Dutch POS tagger that outputs tags from the full D-Coi tag set. A post-processing module was implemented that modifies the output of Frog by introducing Twitter-specific tags where required. In some cases, the output needs to be retokenized as UCTO, the tokenizer used in Frog, introduces too many token boundaries. In particular, retokenization is sometimes required for hashtags, @ mentions, URLs and emoticons. The discourse marker tags can be introduced automatically using regular expressions.

[12] `http://mmax2.sourceforge.net/`

[13] `http://gate.ac.uk/`

[14] Afterwards, it was found that this is a setting that can be changed.

[15] `http://www.comscore.com/Press_Events/Press_Releases/2011/4/The_Netherlands_Ranks_number_one_Worldwide_in_Penetration_for_Twitter_and_LinkedIn`

The SoNaR project provided a dataset containing Dutch tweets. These tweets were all tagged using the POS tagger and the output of the POS tagger was manually corrected by three annotators. This is done with high inter-annotator agreement. This manually corrected sub-set of the Twitter corpus served as the Gold standard for the evaluation of the POS tagger. The results of our experiment indicated that it is possible to automatically annotate tweets, including Twitter-specific tags.

While analyzing the tweets provided by the SoNaR project, we noticed that the tweets from SoNaR corpus contains a significantly different number of correct Dutch words (types and tokens) compared to tweets that were randomly selected. This suggests that the language used in tweets from the SoNaR corpus is different from the language used in random tweets. For the evaluation of our system, this might mean that on random tweets, the performance may be lower than presented here.

There are several ways the POS tagger may be improved. Firstly, the version of Frog used in this research has trouble identifying URLs correctly. This has been resolved within the post-processing module. Another approach might have been to use the meta-data provided with the tweets, which contains information on URLs present in the tweet. Secondly, Frog has a hard time dealing with emoticons. We used a static list of regular expressions of emoticons that the post-processing module could use to recognize them. However, we found that emoticons are used creatively, so using a dynamic method to recognize emoticons might be more effective.

In the meantime, UCTO, the tokenizer used in Frog, has been extended with a Twitter setting that also recognizes URLs, hashtags, @ mentions and emoticons correctly. However, the POS tagger in Frog does not yet recognize Twitter-specific tags and, as such, will not be able to assign these tags.

Another improvement would be the development of a more user-friendly version of the current system. This can be done, for instance, by providing the post-processing module to work on the output of Frog. Alternatively, the Twitter-specific component could be incorporated in Frog itself. This alternative is more attractive, as UCTO can already handle Twitter-specific tokens which are also recognized as such, although currently this specific information is mapped onto more generic token types. This requires retraining of the MBT module in Frog, to allow it to assign Twitter-specific tags.

The project described in this article also served as an educational experiment. Students who followed the Master's course natural language processing could voluntarily participate in the project. The project provided the students a hands-on experience with natural language processing tools. This showed them practical limitations of existing techniques. The time restriction (one week) forced them to make (sometimes sub-optimal) choices to solve the problems at hand. After completing the project, all students were highly enthusiastic about the project, the research, and the presentation of the results.

Acknowledgments

We would like to thank Mehdi Aminian, Zeynep Azar, Alexandros Ntavelos, and Rose Newell for their work during the course of the project. Also, we would like to thank the anonymous reviewers for their useful comments.

References

Aminian, Mehdi, Tetske Avontuur, Zeynep Azar, Iris Balemans, Laura Elshof, Rose Newell, Nanne van Noord, Alexandros Ntavelos, and Menno van Zaanen (2012), Assigning part-of-speech to dutch tweets, *in* Melero, Maite, editor, *Proceedings of the LREC workshop: @NLP can u tag #user_generated_content ?!; Istanbul, Turkey*, pp. 9–14.

Berghmans, J. (1994), *Wotan, een automatisch grammatikale tagger voor het Nederlands*, Master's thesis, University of Nijmegen.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters (2011), *Text Processing with GATE (Version 6).* http://tinyurl.com/gatebook.

Daelemans, W., J. Zavrel, and P. Berck (1996a), Part-of-speech tagging for Dutch with MBT, a memory-based tagger generator, *in* van der Meer, K., editor, *Informatiewetenschap 1996, Wetenschappelijke bijdrage aan de Vierde Interdisciplinaire Onderzoeksconferentie Informatiewetenchap*, TU Delft, The Netherlands, pp. 33–40.

Daelemans, W., J. Zavrel, P. Berck, and S. Gillis (1996b), MBT: A memory-based part of speech tagger generator, *in* Ejerhed, E. and I. Dagan, editors, *Proceedings of the Fourth Workshop on Very Large Corpora*, ACL SIGDAT, pp. 14–27.

Day, D., C. McHenry, R. Kozierok, and L. Riek (2004), Callisto: A configurable annotation workbench, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004); Lisbon, Portugal*, pp. 2073–2076.

Gimpel, Kevin, 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, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers; Portland, OR, USA*, Association for Computational Linguistics, New Brunswick:NJ, USA, pp. 42–47.

Han, B. and T. Baldwin (2011), Lexical normalization of short text messages: makn sens a #twitter, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics; Portland, OR, USA*, Association for Computational Linguistics, New Brunswick:NJ, USA, pp. 368–378.

Kaplan, A.M. and M. Haenlein (2010), Users of the world, unite! the challenges and opportunities of social media, *Business Horizons* **53** (1), pp. 59–68.

Müller, Christoph and Michael Strube (2006), Multi-level annotation of linguistic data with MMAX2, *in* Braun, Sabine, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, Peter Lang, Frankfurt a.M., Germany, pp. 197–214.

Oostdijk, N. and L. Boves (2006), User requirements analysis for the design of a reference corpus of written Dutch., *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006); Genoa, Italy*, pp. 1206–1211.

Oostdijk, N., M. Reynaert, P. Monachesi, G. van Noord, R.J.F. Ordelman, I. Schuurman, and V. Vandeghinste (2008), From D-Coi to SoNaR: A reference corpus for dutch, *Proceedings on the sixth international conference on language resources and evaluation (LREC 2008); Marrakech, Marokko*, ELRA, pp. 1437–1444.

Oostdijk, N., M. Reynaert, V. Hoste, and I. Schuurman (In press), The construction of a 500-million-word reference corpus of contemporary written dutch, *in* Spyns, Peter and Jan Odijk, editors, *Essential Speech and Language Technology for Dutch: Results by the STEVIN-programme*, Springer-Verlag, Berlin Heidelberg, Germany, chapter 13.

Oostdijk, N.H.J. (2011), D-Coi: Dutch language corpus initiative, *in* Renckens, E., editor, *STEVIN Programme Project Results*, Nederlandse Taalunie, pp. 10–11.

Pak, A. and P. Paroubek (2010), Twitter as a corpus for sentiment analysis and opinion mining, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010); Valletta, Malta*, pp. 1320–1326.

Ritter, A., Mausam, and O. Etzioni (2011), A latent dirichlet allocation method for selectional preferences, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics; Uppsala, Sweden*, Association for Computational Linguistics, New Brunswick:NJ, USA, pp. 424–434.

Schmid, Helmut (1994), Probabilistic part-of-speech tagging using decision trees, *Proceedings of the International Conference on New Methods in Language Processing*, pp. 44–49.

Sproat, R., A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards (2001), Normalization of non-standard words, *Computer Speech and Language* **15**, pp. 287–333.

Toutanova, Kristina, Dan Klein, Christopher Manning, and Yoram Singer (2003), Feature-rich part-of-speech tagging with a cyclic dependency network, *Proceedings of the HLT-NAACL; Edmonton, Canada*, North American Chapter of the Association for Computational Linguistics (NAACL), pp. 252–259.

Tromp, E. (2005), *Multilingual sentiment analysis on social media*, Master's thesis, Technical University Eindhoven, Eindhoven, the Netherlands.

Van den Bosch, A., G.J. Busser, W. Daelemans, and S. Canisius (2007), An efficient memory-based morphosyntactic tagger and parser for dutch, *in* van Eynde, F., P. Dirix, I Schuurman, and V. Vandeghinste, editors, *Computational Linguistics in the Netherlands 2007—Selected Papers from the 17th CLIN Meeting; Leuven, Belgium*, pp. 99–114.

Van Eynde, F. (2005), Part of speech tagging en lemmatisering van het D-COI corpus. `http://odur.let.rug.nl/vannoord/Lassy/POS_manual.pdf`.