Dutch factuality classification
Using machine translation to create a Dutch version of FactBank

Harmke Alkemade, s2331357, h.c.alkemade@student.rug.nl,
Jennifer Spenader[*], Johan Bos[†]

January 28, 2016

Abstract

People refer in texts to events that may or may not have happened. Information about how the writer presents an event is called event factuality. Factuality is separated in certainty and polarity. FactBank is an English corpus consisting of events and their corresponding factuality values. There is no such corpus for Dutch, even though this information could be interesting to have. In this project, TechoMT and Google Translate are used to create a Dutch version of FactBank. Sentences are represented by a word vector using frequency information combined with syntactical distance to represent scope. A stochastic gradient learning routine is trained to make a classifier for Dutch. The classifier is tested on a small Dutch corpus consisting of factuality values. The results show that the certainty classification does not perform better than majority-class baseline. Polarity classification however, does perform better. An F-measure of 0.98 is achieved.

1 Introduction

In natural language, people refer to real-word events, which may or may not have happened. It is even possible to refer to an event in which even the speaker doesn't know whether it has happened or not. Information about how the writer presents events mentioned in texts is called event factuality. If an event is factual, it does not necessarily mean that it happened, because the writer can have limited information or present it different from what really happened. Consider the following sentences:

(1) Jack went to soccer training yesterday.

(2) Jack might go to soccer training tomorrow.

In both sentences the author is writing about an event, namely Jack going to soccer training. The first one is presented as corresponding to a fact in the world. The second one however, is presented as a possibility. This makes the first sentence a factual sentence and the second one possibly factual. This information is called *certainty.*

Another aspect of a factuality value is *polarity.* This concept is shown using the following example:

(3) Jack did not go to soccer training yesterday.

The author clearly does not claim that Jack going to soccer training is an event that happened in the real world. The author refers to it as if it certainly did not happen. This is an example of an event with a negative polarity. The events in examples (1) and (2) have positive polarity values. A negative polarity value does not make an event less factual, as the polarity value has no influence on the certainty value. The certainty value of example (3) is CERTAIN, just as example (1).

Current work within Natural Language Processing points out the need for systems to be sensitive to this kind of information. It can help in entailment problems, question-answer systems, information extraction and so on (Saur & Pustejovsky, 2009). Although the factuality value is not necessarily the same as the truth value of an event, combined with reliability of a source it can give a lot of

[*] University of Groningen, Department of Artificial Intelligence

[†] University of Groningen, Computational Linguistics

information. Statements from politicians could for example be compared to factual statements from official reports to check if the claims match.

FactBank (Saur & Pustejovsky, 2009) is a large corpus that consists of annotated factuality values. It contains 208 manual annotated documents, which include all those in TimeBank (Pustejovsky et al., 2006) and a subset of those in the AQUAINT TimeML corpus.* It is only available in English and no comparable corpus exists for Dutch.

Although it would be interesting to have a source for factuality available for Dutch, it takes a lot of effort to make a manually annotated Dutch version of FactBank. Therefore it is worthwhile trying to transfer the information to Dutch in other ways. In this project machine translated versions of FactBank, using Google Translate and TechoMT (Popel & Žabokrtský, 2010), are used to train a stochastic gradient descent classifier. A functioning classifier for Dutch factuality values could be used to make Dutch corpora comparable to FactBank. It could also be used on large scale, to be an extra source of information for information extraction systems. It is unlikely that the machine translated version of FactBank would be suitable to use as a corpus itself, as the the translation is not expected to be completely correct.

An advantage of this method above rule-based systems is that it is easier to transfer to other languages without an extensive research in language specific factuality predictors. It is also easy to extend by training it on a larger dataset instead of humanly analysis on new examples to find out if the rules still work for these new examples.

The research question is: to what extent is it possible to determine certainty and polarity values in Dutch newspaper texts using a machine translated version of the FactBank corpus to train a classifier? It is not a requirement that the translated version of FactBank is completely readable for humans, because it is only used as stepping stone to determine factuality values. The system will be tested against a majority-class baseline, as there is no research found to be comparable enough to use the results as baseline. The results for the certainty and polarity will be considered separately, as the two values have no influence on each other. If this method turns out to be only suitable for one of the two tasks, this can also be detected.

2 Background

2.1 Factuality annotation

Certainty or *modality* expresses the degree of certainty of a source regarding an event. It is a continuum that ranges from truly factual to counterfactual. Speakers are able to map areas of that continuum into discrete values (Lyons, 1977). To make a good transformation to a discrete system, one has to find an expressive enough set of discrete factuality values that is grounded on linguistic intuitions but also supported by commonsense reasoning (Saur & Pustejovsky, 2009). Many linguistics point towards a three-folded distinction, using the values CERTAIN, PROBABLE and POSSIBLE (e.g., Lyons (1977) and Halliday et al. (2014)). These values range from certain to uncertain. The value UNDERSPECIFIED can be used when the source is not clear about the certainty of an event.

Polarity is a binary category that expresses negation. It is divided in POSITIVE and NEGATIVE. Again UNDERSPECIFIED can be used when the source it not clear about the polarity value of an event.

FactBank combines certainty and polarity in one value. Of the 12 different possible values, 8 them are found in FactBank. Events are often represented by a verb. Below are examples of every category that is available in FactBank, in where events are underlined:

(4) Hostels <u>are</u> only 30 percent full. (CERTAIN–POSITIVE)

(5) The company has no <u>estimate</u> of the impact of the earthquake. (CERTAIN–NEGATIVE)

(6) Some had <u>predicted</u> earnings of more than $4 million dollar for this year. (PROBABLE–POSITIVE)

(7) He probably won't <u>finish</u> his report on time. (PROBABLE–NEGATIVE)

(8) Tomorrow the sun may <u>shine</u>. (POSSIBLE–POSITIVE)

(9) It is possible that not my all friends can <u>come</u> to my party. (POSSIBLE–NEGATIVE)

* http://www.timeml.org/site/timebank/timebank.html.

(10) Bella knows whether he likes her. (Certain–underspecified)

(11) John stated that he did not fall. (Underspecified–underspecified)

Certain–underspecified and underspecified–underspecified are the only (partially) uncommitted values. The first can be considered as 'The source knows whether it is the case that X'. Underspecified–underspecified is even less certain: the source does not know what is the factual status of the event, or does not commit to it.

The values that are not in FactBank were selected on beforehand to be not relevant. These values are underspecified positive, underspecified–negative, probable–underspecified and possible–underspecified. To detect possible limitations of the annotation system, participants were allowed to choose one of these values in the annotation task. The values proved to be indeed irrelevant.

To find the factuality value of an event, one has to look at factuality markers (Saur & Pustejovsky, 2009). These are lexical features that can give an indication about the factuality value. Examples of these markers are:

Polarity particles. Polarity particles can often give a good indication of the polarity value of events. They can occur in different lexical parts of a sentence. The interaction of different polarity particles can lead to different outcomes, for example in double negations. Examples of polarity particles are shown in bold in the following sentences:

(12) He did **no** research before he bought his new phone. (Certain–negative)

(13) **None** of them cried during the movie. (Certain–negative)

(14) Elsa is **not** old enough to buy alcohol. (Certain–negative)

Modality particles. Modality particles give information about the certainty value of an event. They can also occur in different lexical parts of a sentence. Their interpretation can be more complex than the polarity particles, because certainty is a range that is divided in three values. Examples of modality particles are shown in bold in the following examples:

(15) **Perhaps** I will stay home tonight. (Possible–positive)

(16) John will **probably** be late again. (Probable–positive)

Event-selecting predicates (ESPs). ESPs introduce a new source and event. The meaning of the ESP influences the factuality value of the introduced event. ESP's are often found in a syntactically that-structure An example is the following sentence:

(17) He **suggested** that his friend should be more kind to his sister.

Discourse structure. It is also possible that one event is presented in one sentence different from the same event in another sentence. My method won't take this into account, as it only looks at syntactic structure of one sentence to classify all events in it.

Events can have different factuality values regarding different sources. The author of the text is a perspective that is always present, but the author could introduce different sources that tell something different about events.

2.2 Previous approaches

One way to automatically determine factuality is to make a rule-based system, such as Saurí & Pustejovsky (2012). The algorithm is a combination of an event recognition system and a factuality classifier. To classify factualities, it looks for syntactic and lexical information in a sentence. Saurí & Pustejovsky (2012) validated their algorithm on the FactBank corpus and consider the certainty and polarity value as one, the same way as FactBank does. The algorithms shows promising results, as it does two tasks in one and it performs much better than the SVM they use as a baseline.

The SVM system that Saurí & Pustejovsky (2012) used as a baseline is inspired on the research of Diab et al. (2009). Diab et al. (2009) report on *believe tagging*, which in many ways is similar to factuality classification. They classify beliefs in the categories Committed belief, Non-committed belief and Non applicable. They also use both lexical and syntactical information for which they use a dependency parser. Instead of a variant of a bag-of-words model, features like 'Part of speech

Table 1: Contingency matrix for distribution of factuality values found in FactBank

	Positive	Negative	Underspecified	Total
Certain	7749	443	12	8204
Probable	363	56	-	419
Possible	226	14	-	240
Underspecified	-	-	4607	4607
Total	8338	513	4619	13470

tag', 'Parent's part of speech tag' and 'Am I a VB with a daughter *to*' are used. Their best feature combination achieves an F-measure of 64%, a relative reduction in F-measure error of 21% over not using syntactic features.

More comparable to my method in the way sentences are represented is Velldal & Read (2012). They used an SVM classifier to identify negated events. They used a bag-of-words model that included information like part of speech, lemmas and forms. Both unigrams and bigrams of variable sizes were considered. The experiments show substantial improvements over the majority-class baseline. They achieve an F-measure of 90 using their classifier, which is an substantial improvement from the majority-class baseline.

3 Method

3.1 Datasets

FactBank (Saur & Pustejovsky, 2009) is used as source to make the training set. The distribution of the different factuality values in FactBank can be found in Table 1. As can be seen, CERTAIN–POSITIVE and UNDERSPECIFIED–UNDERSPECIFIED are far more frequent found than the other values. The frequency of CERTAIN–POSITIVE is not unexpected in news reports, but the frequency of UNDERSPECIFIED–UNDERSPECIFIED is less obvious. This is caused by the fact that in FactBank every level of embedding is annotated. When more than one source of information is involved in a sentence, one event is annotated from the perspectives of these different sources. One of the sources that is always present, is the one of the author. When the author expresses the opinion or perception of a second source (which is not uncommon in newspaper articles) it often omitted whether the author agrees with the second source. Looking at example (11): when John is regarded as source, the factuality value of the event *fall* is CERTAIN–POSITIVE. From the perspective of the author, it is unclear whether he believes John fell or not. In this case, the event is annotated as UNDERSPECIFIED.

The test set will be a set made available for the shared task linked to the CLIN26.† This corpus consists of Dutch translations of 120 English WikiNews articles annotated in a similar way to FactBank. The only difference is that the CLIN dataset has only annotated the inner embedding level, which means that every event is annotated only once. The distribution of the different factuality values in the CLIN dataset can be found in Table 2. As can be seen, the underspecified polarity value is completely left out, mostly caused by the fact that only the inner embedding level is annotated. Remarkable is that two values that are not present in FactBank (UNDERSPECIFIED–POSITIVE and UNDERSPECIFIED–NEGATIVE), are present in the CLIN dataset.

Table 2: Contingency matrix for distribution of factuality values found in CLIN26 factuality dataset

	Positive	Negative	Total
Certain	1047	28	1075
Probable	25	1	26
Possible	29	-	29
Underspecified	27	4	31
Total	1128	33	1161

3.2 Translate FactBank

The first step is to extract all sentences that are included in FactBank, and use them as input for the translation. Translation is done by Google Trans-

†http://wordpress.let.vupr.nl/clin26/shared-task/

late and TechoMT (Popel & Žabokrtský, 2010). The output is a file similar to the one including the English sentences, only a Dutch version. These files are used as input to make a word alignment. This is done using GIZA++ (Och & Ney, 2003). To make the sentences ready for the word alignment, some preprocessing steps are applied on the Dutch and English documents. The first step is to tokenize the sentences. For this I used a script created as a tool for the Europarl corpus (Koehn, 2005), that takes non-breaking prefixes for both languages into account. To enhance the performance of the word alignment from GIZA++, all upper case tokens are replaced with their lowercase counterpart.

The output is a document with three lines of information for each sentence. First a line of meta-information about the length of the target and source sentence and alignment score. The second line is the Dutch translation of a sentence and the last line is the original English version, with brackets between every word. Indexes between the brackets refer to the corresponding Dutch word(s). In some cases there is no word index between the brackets, that means that the word has no equivalent in the Dutch version.

3.3 Aligning factuality values

Another file from FactBank includes the annotations. We need the columns that contain the annotated event, source information for that event, sentence number within that source and the factuality value. The translated events have to be linked to the right factuality value. It starts by iterating over all factuality values from FactBank and storing all events with their factuality value, source and sentence number in a data structure. Then it iterates over the alignment file, and searches for every word in the data structure containing the factuality values. If the word (in combination with the corresponding source and sentence number) is found, the program for the corresponding Dutch word by looking at the index between the brackets and looking for that index in the Dutch sentence and linking the right factuality value to it. When an events has multiple words in the Dutch translation, all words are considered as separate data entries. This is also the place where the factuality value from FactBank is separated in two values; the certainty and the polarity. This method results in a new dataset, containing all information that could be relevant for predicting the factuality value. It has the following columns:

Event. The translated event from FactBank, found between the brackets in the alignment file.

Sentence. The translated sentence from FactBank, using either Google Translate or TechtoMT.

Certainty. Certainty value for the event derived from the first part of the factuality value in FactBank.

Polarity. Polarity value for the event derived from the second part of the factuality value in FactBank.

3.4 Factuality classification

The sentences are represented using a version of the bag-of-words model. After fitting all sentences in a normal word vector, term frequency–inverse document frequency (tf–idf) transform is used to reweight the count features into floating point values suitable for usage by the classifier. This way rare terms are not shadowed by very frequent terms.

The word vector also uses a distance function to represent the syntactical scope of the annotated event. Syntactical information about the sentences is obtained by using Alpino (Bouma et al., 2001), a dependency parser for Dutch. Alpino produces a tree structure of the sentence, which is used to calculate the distance. For every word in a sentence, the syntactical distance is calculated between the event and the word. The value in the word vector is lowered according to this distance. If a sentence contains more than one event, it is found more than once in the dataset. If the syntactical scope would not be represented in the word vector, every event in one sentence would get the the same classification. Including syntactical distance solves this problem as words that appear in the same scope are more important for predicting the factuality value. The following examples shows this:

(18) John did not clean before he left.

Without including syntactical distance, both events would be classified as the same factuality value. A distance function gives *not* a higher value in the entry that represents the event *clean* than for representing *left*.

The syntactical distance is calculated by counting the number of common ancestors of the word and the event. Then for both the word and the event the distance to the closest common ancestor is found by calculating the amount of ancestors and subtracting the amount of common ancestors from it. The syntactical distance is obtained by adding these two values. When the word the and event are in the same scope, the distance value is 0. The further away theword and the event are, the higher the distance value is. Every descend function could be applied in the value, but the function I used is:

$$value = \frac{normValue}{distance^9}$$

If the distance is 0, the value is left unchanged. Various different descent functions were tested but this function gave the best results.

A dataset is created with the previous mentioned word vector and the values for the certainty and polarity. For the learning part Python SciKit learn (Pedregosa et al., 2011) is used. As mentioned in the introduction, the certainty and polarity will be considered separately in the classification task and they also cannot be a feature is determining the other value because that would not be the same as in real situations.

Both tasks also require a different version of the training set, as the CLIN dataset is slightly differently annotated than FactBank. Firstly the CLIN dataset has only annotated the most inner embedded source level. This means that every event is annotated only once. FactBank has annotated all levels, which results in a high amount of UNDERSPECIFIED events. For the certainty task, a version of the training set is used that only consists annotations of the most embedded level. Only considering the most embedded level, the data of FactBank still contains a lot of events that are classified with an UNDERSPECIFIED polarity. CLIN does not include any UNDERSPECIFIED values, so they are left out of the training set.

The word vector is used as training set for a stochastic gradient decent learning routine. For both certainty and polarity classification, various configurations were tested to find the best performing method. Because the data is very skewed, the 'balanced' mode of the algorithm is used, which adjusts the weights for every class x as:

$$weight_x = \frac{n\_samples}{n\_classes \times count(x)}$$

An alpha of 0.001 was used, an epsilon of 0.1 and a squared hinge loss function.

To use the CLIN26 dataset as test set, the data is first transfered to the same notation we used for the training set (event, sentence, polarity and certainty in separate columns). Then directly after creating the word vector for the training set and calculating the distances, the test set is created by using the same word vector. The values in the word vector are calculated using the same descend function as for the training set.

4 Results

4.1 Certainty

The classifier is first tested using 5-fold cross-validation using the different translations as training data. The results of the cross-validation for the TechoMT translation algorithm can be found in Table 3. Even though the weights of the classes are balanced so that the chance is reduced that samples are predicted as the most common category if a lot of examples are classified as CERTAIN. Both the precision and recall of certain are the best, and for all other categories both values are below 0.5.

The results of the Google Translate dataset using 5-fold cross-validation are found in Table 4. The results show that this version performs worse in making a distinction between CERTAIN and UNDERSPECIFIED instances. Recall for UNDERSPECIFIED is lowered from 0.15 to 0.02 and it seems like most of these instances have moved to the CERTAIN category. This lowered the precision for CERTAIN, even though the recall is higher. The Google Translate version performs better in classifying POSSIBLE instances, as both the precision and recall are better for this category.

In Table 5, the results are shown when the TechoMT translation is used as training set and CLIN as test set. CERTAIN is still the class with the highest precision and recall. The results for UNDERSPECIFIED are the most degraded. Almost all instances of this class are classified as CERTAIN.

In Table 6, the results for the Google Translate training set tested on the CLIN dataset are shown. They are slightly better, as for both CERTAIN and POSSIBLE, more instances are classified correctly.

Table 7 shows the results of the classifier that

Table 3: Contingency matrix for certainty classification for 5-fold cross-validation using TechoMT dataset

	Certain	Probable	Possible	Underspecified	Total	Recall
Certain	2715	81	38	180	3014	0.90
Probable	133	49	2	10	194	0.25
Possible	214	15	18	58	305	0.06
Underspecified	1139	42	31	228	1440	0.16
Total	4201	187	89	476	4953	
Precision	0.65	0.24	0.20	0.48		

Table 4: Contingency matrix for certainty classification for 5-fold cross-validation using Google Translate dataset

	Certain	Probable	Possible	Underspecified	Total	Recall
Certain	3041	63	65	43	3212	0.95
Probable	157	41	1	5	194	0.20
Possible	280	13	40	6	305	0.12
Underspecified	1451	51	55	25	1582	0.02
Total	4929	168	161	79	5293	
Precision	0.61	0.24	0.25	0.32		

used both Google Translate and TechoMT as training set. For this version of the classifier, the method is completely the same but the training set consists of both the Google Translate and the TechoMT translations of FactBank. It would be misleading to apply cross-validation on this version of the classifier, because each sentence is added twice and if one of the two is used for training and the other for testing the results can come out positively due to overfitting. The results are not better than for the separate training sets.

An overview of the results of the different methods used to classify the certainty values in the CLIN dataset are found in Table 8. No version of the training set performs better than the baseline.

4.2 Polarity

To classify polarity values, the same classifier is trained. The same training sets are used, but only annotations for the most embedded source are used. As the polarity classification is a different task, configurations were separately tested, but the same configurations showed the best results. The only difference is that the polarity classification has better results when very few features are selected as predictors. Using 10 features gave the best results, as less than 10 features are too few to find good results, but more features leads to less accurate results because the number of POSITIVE instances classified as NEGATIVE increases more than the number of NEGATIVE instance classified as POSITIVE.

Again the performance of the two training sets are first explored using 5-folded cross-validation. The results of the classifier that uses TechoMT translation can be found in Table 9. It performs better in classifying POSITIVE then in classifying NEGATIVE instances, as both precision and recall are higher. Similar to certainty, for polarity the classifier also shows better results when the Google Translate version of the training set is used. The results of this can be found in Table 10.

An overview of the results of the different methods in the CLIN dataset can be found in Table 11. The precision of the methods using Google Translate, TechoMT and the two combined as training set have the same scores. Applying 5-fold cross-validation on the CLIN dataset using the same configurations also gives the same results. They all perform better than the majority-class baseline. The detailed results of the classifiers applied on the CLIN dataset can all be found in Table 12, as they all have the same outcome.

Table 5: Contingency matrix for results on the CLIN dataset using TechoMT training set

	Certain	Probable	Possible	Underspecified	Total	Recall
Certain	920	42	3	97	1072	0.86
Probable	19	8	0	2	29	0.28
Possible	24	1	0	1	26	0.00
Underspecified	24	3	2	2	31	0.06
Total	987	54	5	102	1158	
Precision	0.93	0.15	0.00	0.02		

Table 6: Contingency matrix for certainty classification in CLIN dataset using Google Translate as training set

	Certain	Probable	Possible	Underspecified	Total	Recall
Certain	959	38	26	49	1072	0.93
Probable	20	8	0	1	29	0.28
Possible	18	2	6	0	26	0.23
Underspecified	29	0	0	2	31	0.06
Total	1026	48	32	52	1158	
Precision	0.93	0.17	0.19	0.04		

5 Discussion

The classifier is able to perform better than the baseline for the polarity classification task, for both translations and the combined version. All methods have a slightly better precision than the majority-class baseline in which all instances are classified as CERTAIN, but none of them has a better F-score. Remarkable is that all systems perform equally well.

The certainty classification task does not show better results than the baseline. Although the majority-class baseline is very high because of the distribution of the data, this method does not seem to be suitable for the classification of certainty value. The different translations score differently on the task, which indicates that a better translation would lead to a better success rate.

The fact that polarity classification had better results than certainty classification seems to show that polarity classification is easier than certainty classification. This is also supported by the fact that both translations perform equally well on the polarity classification task. This shows that the quality of the translation is less important. Certainty classification is probably harder because although is it for people easy to make a distinction between the different certainty values, this can be different for a classification system that has a limited amount of examples of different classes.

From most of the results can be seen that Google Translate performs better than the TechoMT translation. By inspecting the translations manually there are some errors that can be noticed. The TechoMT translation algorithm doesn't translate parts between quotes and also leaves written numerical values untranslated. Google Translate seems to look less at the context of a word while translating, which can also causes errors. This makes both translations not perfect, but this was already expected.

Because the translation is the first step in the process, it has also influence on all other steps in the process. The alignment can not perform optimally if the sentences are not translated completely correctly. If the alignment is not correct, wrong words will be selected from the Dutch sentence as the event that is annotated in FactBank. It is also harder for Alpino to parse the sentences if they are not translated completely correct. This can cause errors in the distance calculations.

As the quality of the output of these tools cannot be controlled, it is hard to say what causes the errors in the classification. The biggest factor seems to be the quality of the translation.

For future work it would be interesting to test

Table 7: Contingency matrix for certainty classification in CLIN dataset using Google and TechoMT as training sets

	Certain	Probable	Possible	Underspecified	Total	Recall
Certain	912	46	16	98	1072	0.85
Probable	19	8	0	2	29	0.28
Possible	24	1	0	1	26	0.00
Underspecified	24	3	2	2	31	0.06
Total	979	60	18	103	1158	
Precision	0.93	0.13	0.00	0.19		

Table 8: Performance of classification certainty in CLIN dataset using different training sets

	Precision	Recall	F-scores
Google Translate	0.87	0.84	0.86
TechoMT	0.87	0.81	0.84
GT + TechoMT	0.87	0.84	0.85
Baseline	0.86	0.93	0.89

Table 9: Contingency matrix for polarity classification for 5-fold cross-validation using TechoMT dataset

	Negative	Positive	Total	Recall
Negative	261	193	454	0.57
Positive	868	6153	7021	0.88
Total	1129	6346	7475	
Precision	0.23	0.97		

the method on the original English sentences, before putting the translation step in between. At this moment it is hard to say whether the errors are caused by the quality of the translation or by the method itself. If the errors are caused by the quality of the translation, it could be used for languages that have better machine translation available. There is no point in putting effort in translating FactBank manually, as these efforts could be better put into finding another system to annotate factuality values in Dutch. By trying the method for English it can also be easier to compare with different methods, as there are at the moment not yet other results published on the same dataset. Testing the method on FactBank is also better because the dataset is bigger. The CLIN dataset has the limitations of one annotation per event, but also some categories that are poorly represented by the small amount of examples.

Table 10: Contingency matrix for polarity classification for 5-fold cross-validation using Google Translate dataset

	Negative	Positive	Total	Recall
Negative	222	270	492	0.45
Positive	88	7376	7464	0.99
Total	310	7646	7956	
Precision	0.72	0.96		

Table 11: Performance of polarity classification in CLIN dataset using different training sets

	Precision	Recall	F-scores
Google Translate	0.98	0.97	0.98
TechoMT	0.98	0.97	0.98
GT + TechoMT	0.98	0.97	0.98
Baseline	0.86	0.93	0.89

The results for classifying polarity for this research are higher than the results of Velldal & Read (2012), but because it used another language and test set, it is hard to make the comparison. The CLIN dataset has also a very small number of examples for NEGATIVE polarity, so it is hard to generalize the results on events in general. Because this project uses a completely other way to represent

Table 12: Contingency matrix for polarity classification in CLIN using both TechoMT and Google Translate versions of the training set

	Positive	Negative	Total	Recall
Positive	24	9	33	0.73
Negative	24	1101	1125	0.98
Total	48	1110	1158	
Precision	0.50	0.99		

the data, it could be interesting to compare the two. This could be done by testing this method in English.

This method is not complete in that events have to be selected first before the events can be classified. A complete system would also be able to annotate event factualities of all embedding levels. To make a system that can predict factuality values for all embedding levels, is it important to include the source somewhere in the method. It is probably hard to represent the source in the word vector, so when doing that the representation of the events should change a bit.

Besides the training algorithm and its configurations, the distance function is also an influence factor of the results. Without the distance function all events in one sentence would have the same value. The system was also tried without the distance function, which showed that it performed much better when it is added. Other descent functions could be tried to find which suits best.

It is questionable whether is it important to classify UNDERSPECIFIED instances and make the distinction between PROBABLE and POSSIBLE instances. Diab et al. (2009) do not make that distinction and mention that it is not important for all applications. Although Saur & Pustejovsky (2009) show that human annotators had no confusion about PROBABLE and POSSIBLE, this could be different for this kind of system.

In conclusion, some results are promising, but for the complete task (classifying both certainty and polarity) the method is not sufficient. This could be solved by making the task easier by combining for example POSSIBLE and PROBABLE or testing with other descend functions and configurations. It would be better to test the method on English first, to establish if the method works without having to use a translation of FactBank that is not completely perfect. Another solution could be to combine this method with a rule-based system. A rule-based system could classify certainty values and the polarity values could be classified using this method.

References

Bouma, G., Van Noord, G., & Malouf, R. (2001). Alpino: Wide-coverage computational analysis of dutch. *Language and Computers*, *37*(1), 45–59.

Diab, M. T., Levin, L., Mitamura, T., Rambow, O., Prabhakaran, V., & Guo, W. (2009). Committed belief annotation and tagging. In *Proceedings of the third linguistic annotation workshop (law)* (pp. 68–73).

Halliday, M., Matthiessen, C., Halliday, M., & Matthiessen, C. (2014). *An introduction to functional grammar.* Taylor & Francis.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Mt summit* (Vol. 5, pp. 79–86).

Lyons, J. (1977). *Semantics.* Cambridge University Press.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*(1), 19–51.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Popel, M., & Žabokrtský, Z. (2010). Tectomt: Modular nlp framework. In *Proceedings of the 7th international conference on advances in natural language processing* (pp. 293–304). Berlin, Heidelberg: Springer-Verlag.

Pustejovsky, J., Verhagen, M., Saurí, R., Littman, J., Gaizauskas, R., Katz, G., ... Setzer, A. (2006). Timebank 1.2. *Linguistic Data Consortium*, *40*.

Saurí, R., & Pustejovsky, J. (2012, jun). Are you sure that this happened? assessing the factuality degree of events in text. *Comput. Linguist.*, *38*(2), 261–299.

Saur, R., & Pustejovsky, J. (2009). Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation*, *43*(3), 227-268. doi: 10.1007/s10579-009-9089-9

Velldal, E., & Read, J. (2012). Factuality detection on the cheap: Inferring factuality for increased precision in detecting negated events. In *Proceedings of the workshop on extra-propositional aspects of meaning in computational linguistics* (pp. 28–36). Stroudsburg, PA, USA: Association for Computational Linguistics.