Project Number: **770299**

NewsEye:

A Digital Investigator for Historical Newspapers

Research and Innovation Action
Call H2020-SC-CULT-COOP-2016-2017

D3.6: Stance Detection (final)

Due date of deliverable: M24 (30 April 2020)

Actual submission date: 30 April 2020

Start date of project: 1 May 2018 **Duration:** 36 months

Partner organization name in charge of deliverable: ULR

Project co-funded by the European Commission within Horizon 2020		
Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	-
RE	Restricted to a group specified by the Consortium (including the Commission Services)	-
CO	Confidential, only for members of the Consortium (including the Commission Services)	-

Revision History

Document administrative information		
Project acronym:	NewsEye	
Project number:	770299	
Deliverable number:	D3.6	
Deliverable full title:	Stance Detection (final)	
Deliverable short title:	Stance Detection (final)	
Document identifier:	NewsEye-T32-D36-StanceDetection-final-Submitted-v3.0	
Lead partner short name:	ULR	
Report version:	V3.0	
Report preparation date:	30.04.2020	
Dissemination level:	PU	
Nature:	Report	
Lead author:	Ahmed Hamdi (ULR), Thi Tuyet Hai Nguyen (ULR)	
Co-authors:	Antoine Doucet (ULR)	
Internal reviewers:	Mark Granroth-Wilding (UH-CS), Jani Marjanen (UH-DH)	
Status:		Draft
		Final
	x	Submitted

The NewsEye Consortium partner responsible for this deliverable has addressed all comments received, making changes as necessary. Changes to this document are detailed in the change log table below.

Change Log

Date	Version	Editor	Summary of changes made
16/03/2020	0.1	Ahmed Hamdi and Thi Tuyet Hai Nguyen (ULR)	First complete version of the document
23/03/2020	0.2	Antoine Doucet (ULR)	Adjustments and reorganisation.
26/03/2020	1.0	Ahmed Hamdi and Thi Tuyet Hai Nguyen (ULR)	First full draft.
01/04/2020	1.1	Mark Granroth-Wilding (UH-CS) and Jani Marjanen (UH-DH)	Internal review including minor edits.
06/04/2020	1.2	Ahmed Hamdi, Thi Tuyet Hai Nguyen and Antoine Doucet (ULR)	Internal reviews taken into account.
10/04/2020	2.0	Antoine Doucet (ULR)	Additional proofreading leading to final version.
23/04/2020	2.1	Ahmed Hamdi, Thi Tuyet Hai Nguyen and Antoine Doucet (ULR)	Extended result analysis and quality management feedback taken into account.
30/04/2020	3.0	Antoine Doucet (ULR)	Final modifications towards submission.

Executive summary

The present deliverable is the final version of our work on stance detection. A first Deliverable D3.3 was focused on the prior state of the art and delivered at M12. This final report, delivered at M24, is describing robust-to-noise and language-independent approaches to stance detection.

Part of WP3, Task T3.2 of the NewsEye project deals with stance detection. Stance detection is the task of determining whether the author of a piece of text is in favor of a given target, against a given target, or neutral. It is a subtask of sentiment analysis where opinion is not expressed in general but with respect to a specific target. This task immediately follows Task T3.1 on named entity recognition (NER) and linking (NEL), and is in charge of attaching stance to the NEs as recognised in T3.1, and possibly also to the topics determined in T4.1. Task T3.2 will provide data usable both directly by end-users through the NewsEye demonstrator, and as input to other analyses described in WP4.

The analysis of the state of the art showed the lack of appropriate resources and tools for the evaluation of stance detection as defined in the NewsEye project, in particular in the context of historical newspapers. Therefore, the works led and presented in the present report include the development of new datasets and methods to perform and evaluate the task of stance detection within the NewsEye project.

This report presents three methods[1] developed in the context of this deliverable, which rely on sentiment lexicons and machine learning. In addition, in collaboration with Task T1.3 (data generation) running until M36, we launched the construction of a dataset for training and testing over real NewsEye project data. The results reported in our experiments involve this novel resource, as well as larger external datasets, which are not strictly matching our definition of the stance detection task and have no ATR errors, yet are close enough to our goals to shed relevant light on the methods we developed.

[1] Publicly available at https://github.com/NewsEye/Stance-Detection

Contents

1 Introduction

Stance detection is one of the active subareas of natural language processing (NLP), aiming to identify the stance of an author (i.e. favour, against, and neutral) towards a target entity (e.g., person, organisation, etc.) that is either explicitly mentioned or implied in the text. Some initial works on stance detection concentrate on identifying stance in online debate forums [1, 2]. Recently, a competition on detecting stances from tweets was organised in the annual Workshop on Semantic Evaluation (SemEval-2016) [3]. Another shared task (FNC-1) on fake news stance detection was performed in 2017[2]. In this task, stance is the relationship between a headline and a body text from a news article, which can be one of the following four options: agrees, disagress, discusses, or unrelated. Stance detection was also explored in rumour stance classification, which categorises the stance of posts into supporting, denying, querying or commenting [4]. In this context, a large part of data on the stance detection task belongs to the domains of social media, news, and user comments.

In the context of NewsEye, we concentrate on detecting stances in historical documents. Our stance detection methods judge whether the body text of a news article is positive, negative, or neutral towards a given named entity mentioned in the text. In other words, the task can be considered as a classification problem that categorises two pieces of text, i.e. the article and the target entity, into three classes of stance (positive, negative and neutral).

There are some challenges to determine the stance towards a target entity. First, the stance on named entities may be expressed later in the context of corresponding pronouns or abbreviations. Systems should therefore be able to correctly identify the named entity to which pronouns refer. Second, the texts from which stances are detected are coming from historical digitised newspapers that may be noisy due to errors caused by automated text recognition (ATR). Furthermore, historical texts often differ from modern texts in their spellings, which can result in low performance, for instance if contemporary polarity lexicons are used to get polarity scores of historical words.

In the first report on this task, Deliverable D3.3, we described state of the art techniques dealing with stance detection. However, very few systems are adequate for stance detection as defined in NewsEye and these systems are rarely publicly available, as we have detailed in D3.3. In this deliverable, we learned from the existing methods in order to implement our own methods with respect to NewsEye constraints. This work reports experimental results of different stance detection methods for better understanding of their drawbacks and benefits. There are two main differences between existing works (most of them described in D3.3) and methods implemented in this work. First, target entities are named entities that are necessarily mentioned in the text. So they have to be considered to classify the body text into stance classes. Second, the body text may contain several named entities, possibly each with a different stance; the classification task should therefore take into account each of the named entities separately.

This deliverable is to determine the most adequate method for stance detection, to be applied over the whole NewsEye dataset. The resulting automated annotations will then be integrated to the NewsEye demonstrator. Stance annotations will thus be accessible in two ways. First, users will be able to view them directly when accessing newspapers through the NewsEye platform[3]. Second, they will be accessible to developers through APIs, notably used by the personal research assistant developed in WP5 and the dynamic analysis tools developed in WP4.

[2] http://www.fakenewschallenge.org/, accessed on 12/03/2020
[3] https://platform.newseye.eu/

The deliverable is structured as follows: an overview of related works is provided in Section 2 while the details on our methods are given in Section 3. Section 4 describes the used evaluation corpora, including the novel dataset developed in the project. Experimental results are described and discussed in Section 5, before we conclude this report in Section 6.

2 State of the art

This section gives a brief and updated overview of the state of the art developed in Deliverable D3.3.

There have been many research studies on stance detection in the last few years [5]. Many of them have been proposed in the SemEval-2016 shared task on Stance Detection in Twitter [3]. Works on stance detection can be categorised into two main approaches: **lexicon-based approach** which considers that sentiment information of a piece of text strongly impacts the stance and **machine learning (ML) approach** which includes both feature-based ML techniques and more recently deep learning ones.

The first approach generally extracts sentiment information relying on a lexicon of words that designate a sentiment, such as SentiWordNet [6]. Normally, each lexicon word is associated with a set of opinion scores and polarity scores. These scores are used to calculate the polarity score of a given text. The simplest lexicon-based method consists of assigning a text a score equal to the total number of words that contain an opinion present in the document [7, 8].

Although several sentiment lexicons are available in English, such a resource is still missing for most languages. In order to deal with the multilingual sentiment detection problem, some approaches have been proposed. Denecke [9] suggests sentiment analysis based on SentiWordNet. Particularly, the author translates non-English documents into English. Next, the polarity of each translated word is extracted from SentiWordNet.

Instead of translating the source languages into English, Chen et al. [10] build high-quality sentiment lexicons for 136 major languages (e.g., English, French, German, Finnish, etc.) by producing and using a large knowledge graph. These lexicons obtain a polarity agreement of 95.7% with published lexicons. For each language, the authors provide two-word lists, i.e. a list of positive words, and a list of negative words. Unlike SentiWordNet, there are no polarity scores in these word lists.

The second approach includes ML techniques. It consists in providing data to a classifier in order to generate a model that is used for the test portion of data. This type of approach has two aspects: the extraction of features and the training of the classifier. Stance detection using machine learning relies on a human-annotated training corpus. It can be seen as a text classification problem with three continuous classes. While text classification methods aim to label pieces of text with a set of predefined classes, the stance detection task categorises a piece of text towards a target entity.

The main features used are unigrams, bigrams, trigrams [11], parts-of-speech tags, similarities and polarities [12], as well as word and character embeddings [13]. Once all the features are extracted, they are provided to classifiers such as Suppport Vector Machine (SVM) [12] and Naive Bayes [11] as well as majority vote classifiers [14] in order to classify them into three classes: positive, negative, and neutral.

More recently, deep learning techniques are used for extracting stances toward target entities from a

Language	# positive words	# negative words
English	1,421	2,955
Finnish	1,333	1,962
French	1,615	3,038
German	1,510	2,464
Swedish	1,456	2,266

Table 1: Number of polarity words for English, German, Finnish, French and Swedish.

piece of text [15, 16, 17]. Augenstein et al. [18] apply bidirectional Long Short Term Memory networks (bi-LSTMs) in order to encode the text and the target entity, then, a softmax activation function is applied to predict the stance of the target-text pair. Zarella et al. [19] use word2vec embeddings [20] in order to convert words into feature vectors. Then, they learn sentence representations using these vectors. Finally, the sentence vectors are fine-tuned for stance detection. In a similar setting, Pivovarova et al. [21] train word vectors using the Global Vectors for Word Representation (GloVe) algorithm [22]. Language models based on the transformer architecture are also explored to achieve improvements on fake news challenge stance detection [23]. In this task of fake news detection, authors determine the stance of a news article relative to another news article. The stance can either agree, disagree, discuss the same topic, or be completely unrelated.

In this work, we developed and tested several methods from the two approaches which can perform with small-data and large-data scenarios. Our lexicon-based methods are similar to those of Chen et al. [10]. We take advantage of the multilingual lexicons published to detect the stance of documents written in the NewsEye languages i.e., Finnish, French, German, and Swedish. Regarding the machine learning approaches, our methods rely on several supervised classifiers, using text features of different word representation methods such as learning word embedding methods. We additionally employ a deep learning method to automatically extract stances.

3 Description of the methodology

In order to produce an exhaustive study of stance detection, we developed and tested several methods from the two types of approaches of the state of the art (cf. Section 2). In this section, we first describe the lexicon-based approach, followed by the machine learning methods.

3.1 Lexicon-based approach

A lexicon-based method relies on opinion-bearing words (or opinion words) that are commonly used to express sentiments (i.e. positive, negative or neutral). To detect stance, our method employs sentiment lexicons from the state of the art [10], which are however provided without polarity scores. The NewsEye project initially focuses on processing historical newspapers written in Finnish, Swedish, French, and German. Therefore, this report will focus on sentiment lexicons for these languages. Table 1 indicates the number of negative and positive words for each of these languages.

The detail of our approach and an example are shown in Figure 1. Firstly, after removing stop words, numbers, and punctuation, we collect the words neighbouring the target within a window of size $2 \times n$ (n previous words before, n next words). Next, the number of positive and negative opinion words within that window are used to compute the average polarity score, as the number of positive/negative words

divided by the total number of opinion words. The average numbers are rounded to two decimal places and are applied to determine the sentiment of the given target. Particularly, if the absolute value of the difference between the negative score and the positive score is less than or equal to a threshold k, then the stance is classified as neutral. If the negative score is larger than the positive one, then the stance is negative. Otherwise, it is positive. Both n and k are selected depending on each dataset.

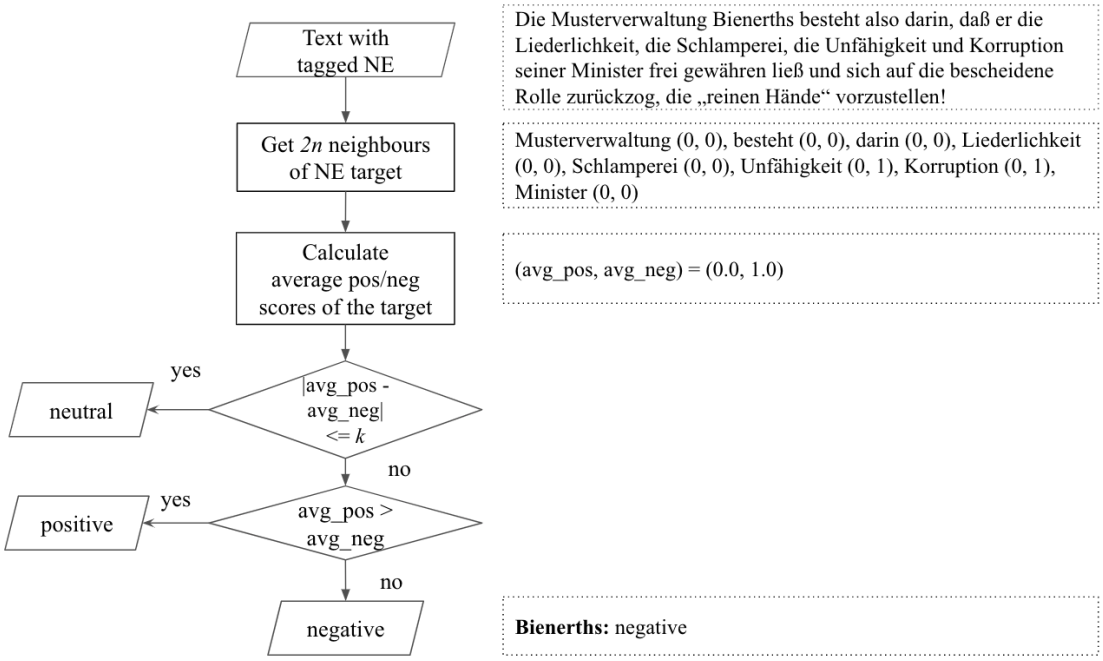

Figure 1: The steps of the lexicon-based approach and an illustrative example in German. The English translation of the text is "*The model administration of Bienerth consists in the fact that he allowed the licentiousness, the sloppiness, the incompetence and the corruption of his ministers to be freely granted and he withdrew himself to the modest role of presenting 'clean hands'!*". The only words in the window ($n = 7$) that belong to the German sentiment lexicon are “Unfähigkeit” and “Korruption”, bearing negative sentiment.

3.2 Machine learning approach

Machine learning methods rely on corpora being annotated by experts in order to train models that will be used to predict stances. In this section, we propose two methods, 1) a feature-based ML method relying on a majority vote classifier from the predictions of nine trained classifiers and 2) a deep learning method based on a multi-layer bidirectional encoder.

3.2.1 Feature-based ML approach

The feature-based methods are carried out in two stage. First, the target entity as well as the body text are converted into feature vectors. Second, a classifier uses annotations in order to classify the stance into three classes: positive, negative, and neutral.

In this work, we trained 9 supervised classifiers for stance detection toward named entities: logistic regression (LR), LR optimised by the stochastic gradient descent (SGD), K-nearest neighbours, support vector machine (SVM), random forest (RF), Adaboost classifier, decision tree (DT), linear discriminant

analysis (LDA) and Gaussian naive Bayes (Bayes). Each classifier takes as input a vector feature encoding the body text and the target, and generates a stance for each of the input pairs. Predictions from these nine trained classifiers allow selecting a single class via a majority vote.

In order to convert targets and body texts into vectors, we initially used FastText[4] pre-trained word vectors. FastText embeddings provide a mapping of 2 billion words to a 300-dimensional vector, pre-trained on Wikipedia data for 157 different languages [24]. It is thus a very good match with respect to NewsEye's ambition to process documents in any language. FastText uses both character-based and subword-based embeddings to deal with out-of-vocabulary (OOV) words [25]. OOV words can be frequent in historical texts due to language changes, spelling variations and OCR errors. Word vectors are then weighted using the *tf-idf* measure. We calculated for each word its term frequency (*tf*) and its document[5] frequency (*df*), which is the number of documents in which the word appears. The inverse document frequency (*idf*) score of each word is calculated using this function:

$$idf_{word} = ln(\frac{1 + N(number\ of\ docs\ in\ the\ collection)}{1 + df_{word}}) + 1 \quad (1)$$

The *tf-idf* values of each word in a document are computed by the multiplication of the term frequency and the inverse document frequency. Each document (target, body text) is then converted into a FastText vector by multiplying the *tf-idf* score of each word in the document by the FastText vector of the word, which is finally summed up and normalised according to the document length.

Target entities are NEs that can be replaced by pronouns, abbreviations, or even simply altered by the ATR process. In addition, the news article may have many mentions of the same named entity. In order to treat potential dependencies that could exist among related targets, we enriched the vector representations of the documents by two additional features: the cosine similarity between the news article (body text) and the target entity as well as the number of common n-grams between them.

1. the cosine similarity between the target (X) and the body text (Y) of each document.

$$cosine\_sim(X, Y) = \frac{\sum_{i=0}^{300}(X_i \times Y_i)}{\sqrt{\sum_{i=0}^{300}(X_i)^2} \times \sqrt{\sum_{i=0}^{300}(Y_i)^2}} \quad (2)$$

   where X and Y are respectively the 300-dimensional vector representations of the target entity and the body text. Since NEs can appear in different ways in the body text, it is essential to distinguish all the mentions referring to the NE. The cosine similarity measure gathers all the mentions that have similar meanings.
2. the number of common n-grams in the target entity and the body text. It is the number of n words in the target that occur in the body text. The n-grams used in this work are uni-grams, bi-grams and tri-grams. As we mentioned in the introduction, the target entity is always mentioned in the text; however, sometimes the body text does not contain the whole entity. For instance, in the following example, only one word of the entity appears in the body text.
   - body text: *I have no problem with Ronaldo and I shook hands with him at the end.*
   - target entity: *Ronaldo Luis Nazário de Lima*
   - stance: positive

[4] https://fasttext.cc/docs/en/crawl-vectors.html, accessed on 12 March
[5] In this work, a document includes the target and the body text pair.

3.2.2 Deep learning approach

Bidirectional encoder representations from transformers (BERT) [26] is a well-known contextual language representation model, notably because the pre-trained BERT model can be fine-tuned to handle a variety of downstream tasks. In this work, we investigate the application of BERT model on stance detection. BERT is a multi-layer bidirectional transformer encoder. Transformer is an encoder-decoder structure with multi-head attention mechanisms. This model adds positional encodings to the input embeddings to inform the model about the sequence order instead of using recurrence or convolution like typical encoder-decoder models.

BERT is pre-trained on unlabelled data over two different tasks: Masked Language Model (MLM), and Next Sentence Prediction (NSP). In the MLM task, the authors obtain a bidirectional pre-trained model by randomly masking some percentages of the input tokens, and predicting those masked tokens. The second task (NSP) enables the model to learn the relationship between two sentences.

The BERT authors mainly present results on two model sizes including BERT-base (L=12, H=768, A=12), and BERT-large (L=24, H=1,024, A=16) with L as the number of transformer blocks, H as the size of hidden layer, A as the number of self-attention heads. They outperform state-of-the-art of several NLP tasks by fine-tuning their pre-trained BERT models. The fine-tuned model is firstly set with pre-trained parameters that are then adjusted according to the input and output data of downstream tasks.

There are multiple task-specific BERT models, some of them work at sentence-level, others perform at token-level. The BERT models showed an ability to process historical OCRed data extracted from newspaper articles in many NLP tasks such as named entity recognition [27] and the analysis of news articles [28]. In our case, stance analysis toward given named entities might be considered as a sentence pair classification task as illustrated in Figure 2. The first sentence is the body text, the second one is the given named entity, and the class label consists of positive, negative or neutral values. Similar to other fine-tuned models, the two sentences are tokenized by WordPiece [29]. Next, they are packed together into a single pair of sequences along with special classification tokens (i.e. *[CLS]* at the beginning of the sequence, and *[SEP]* at the end of each sentence).

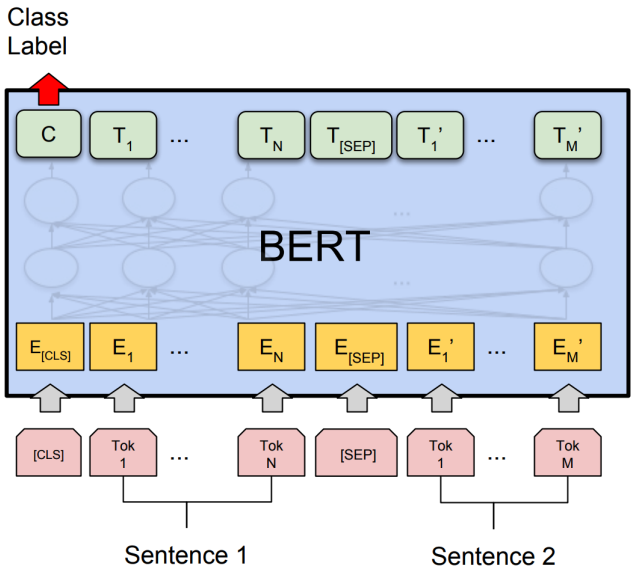

Figure 2: BERT for sentence pair classification [26].

Our fine-tuning method is similar to Valeriya's approach [23] that classifies stances of a body text towards a claim mentioned in a target entity. We experimented with both the cased and uncased BERT-base models. Typically, the multilingual model is employed. However, if a BERT model of a specific language is available, this model will be selected.

4 Evaluation datasets

In order to evaluate our methods, we conducted experiments over several benchmarks. In this section, we describe the NewsEye dataset created within the project as well as other datasets we used to enrich and assess our approaches.

4.1 NewsEye dataset

As we mentioned above, the present task aims to detect the stance towards named entities that are recognised from full text. However, most of the existing datasets are not adapted to the NewsEye context. First of all, in many datasets, the stance is annotated towards entities that are not necessarily mentioned in the text [30, 31, 32]. Second, to the best of our knowledge, there is no prior work attempting to detect stance over historical documents.

Therefore, in order to evaluate the performance of our approaches in the context of historical newspapers in several languages, we built a dataset within the project, using the historical newspapers of the partner libraries. The annotation produced combined named entity recognition, linking, and stance, resulting in the NewsEye NE dataset v1, further described in Deliverable D3.5 on named entity recognition and linking.

Four resources have been provided, corresponding to the 4 languages of the project: de-NewsEye, fi-NewsEye, fr-NewsEye and sv-NewsEye, for German, Finnish, French and Swedish, respectively. In all of them, named entities were extracted by human experts. Named entities cover people, locations, organisations, and media products such as newspapers, magazines, broadcasts, etc. For each named entity, a stance is annotated to mark the opinion of the author toward this entity. Examples are shown in Table 2. In the first column, we give an extract of the news articles followed by their "*English translations*".

Unsurprisingly, neutral stance dominates the other stance types with around 98% of the annotations on average. In de-NewsEye, for example, 8,845 stances are annotated, among which only 99 are positive and 166 are negative. While this has no impact on the effectiveness of lexicon-based approaches, the small number of positive and negative stances is an issue for supervised learning methods. Table 3 summarises the NewsEye dataset v1 used in this work. It shows the number of stances annotated and their distribution according to each class: negative (NEG), positive (POS), neutral (NEUT).

Inter-annotator agreement (IAA). In order to evaluate the IAA in the NewsEye dataset, several identical pages of each corpus have been annotated in parallel by two groups of experts, native speakers of the corresponding languages. The distribution of stance annotations between the two groups is detailed in Table 4. Each cell indicates the number of cases when the 1^{st} and 2^{nd} group respectively annotated an NE with the stance class in the row and column, respectively. For each dataset, the IAA is calculated using the Kappa coefficient introduced by Cohen [33]. Cohen's Kappa measures the agreement

Text fragment	Named entity	Stance	Language
Die Musterverwaltung Bienerths besteht also darin, daß er die Liederlichkeit, die Schlamperei, die Unfähigkeit und Korruption seiner Minister frei gewähren ließ und sich auf die bescheidene Rolle zurückzog, die „reinen Hände“ vorzustellen! "*The model administration of Bienerth consists in the fact that he allowed the licentiousness, the sloppiness, the incompetence and the corruption of his ministers to be freely granted and he withdrew himself to the modest role of presenting 'clean hands'!*"	Bienerths	NEG	German
Päinwastoin teki kapellimestari Wigna nytkin mitä tarkinta ja inspireeratuinta työtä "*Even then, conductor Wigna did the most accurate and inspired work*"	kapellimestari Wigna	POS	Finnish
Briand communiqua ses propositions au ministre des Affaires Etrangères Ribot. "*Briand communicated his proposals to the Minister of Foreign Affairs Ribot.*"	Briand	NEUT	French

Table 2: Example annotations from the NewsEye dataset.

	POS	NEG	NEUT	Total
de-NewsEye	99	166	8,580	8,845
fi-NewsEye	54	40	2,575	2,669
fr-NewsEye	44	24	10,259	10,327
sv-NewsEye	51	10	2,383	2,444

Table 3: Number and distribution of annotated stances in the NewsEye dataset v1.

between two annotators, while taking into account the possibility of chance agreement:

$$IAA = \frac{p_a - p_o}{1 - p_o} \tag{3}$$

where p_a is the relative observed agreement among annotators, and p_o is the hypothetical probability of chance agreement.

The distribution highlights that positive and negative stances are rare, and that the agreement about them is very low. Indeed, few positive and negative stances are annotated in agreement by the annotators. However, we also observe that there is no single case when one annotator considers positive something that another annotator considers negative. We believe that this highlights the large scale that exists between a fully neutral and fully non-neutral stance, especially as expressed in the context of newspapers. At the same time, seeing no strong disagreement in the annotations (for instance, positive vs. negative) confirms the applicability of stance annotations in the context of big data, for quantitative rather than fine-grained analysis.

4.2 Overview of external datasets

Additionally to NewsEye data, in order to get a more extensive evaluation of our approaches, we used two available English datasets with a balanced distribution of stances: the EMM corpus [34] and the

fr-NewsEye IAA = 0.80		2^{nd} group			
		POS	NEG	NEUT	Total
1^{st} group	POS	**2**	0	0	2
	NEG	0	**0**	0	0
	NEUT	1	0	**125**	126
	Total	3	0	125	128

de-NewsEye IAA = 0.23		2^{nd} group			
		POS	NEG	NEUT	Total
1^{st} group	POS	**1**	0	3	4
	NEG	0	**1**	7	8
	NEUT	1	1	**132**	134
	Total	2	2	142	146

fi-NewsEye IAA = 0.48		2^{nd} group			
		POS	NEG	NEUT	Total
1^{st} group	POS	**4**	0	3	7
	NEG	0	**0**	0	0
	NEUT	4	0	**62**	66
	Total	8	0	65	73

sv-NewsEye IAA = 0		2^{nd} group			
		POS	NEG	NEUT	Total
1^{st} group	POS	**0**	0	0	0
	NEG	0	**0**	0	0
	NEUT	1	0	**13**	14
	Total	1	0	13	14

Table 4: Distribution of annotations according to the stance classes in the NewsEye data.

PULS corpus [21]. We choose these datasets for two main reasons. First, they have similarities to the NewsEye context since they are based on (contemporary) news articles and broadcasts and the stances are on people and organisations (which are some of the types included in the NewsEye named entities). Second, both of these datasets have high inter-annotator agreement.

The EMM corpus consists of a collection of 1,592 quotations extracted from English newspaper articles in April 2008. A quotation is a short reported speech where the source is known. Authors assume that quotations are usually more subjective than the other parts of news articles. Each quotation is associated with:

- Source name: the name of the person who has made the statement.
- Target Name: the target entity of the quotation.
- Annotations: polarities given by annotators. For each quotation, 2 annotators among the 4 are asked to annotate the entity mentions. In the cases where the 2 annotators disagree, the third annotator decides in order to reach a gold standard annotation. The polar judgements (positive or negative) towards target entities are estimated from the author's text and detected independently of the opinion held by annotators.
- Agreement: attribute indicates whether there is an agreement between annotators.

The result was a corpus of 1,592 quotations. The inter-annotator agreement for this dataset reaches 81%. These quotations are categorised into 234 (15%) negative quotations toward target entities, 193 (12%) positive and 895 neutral quotations which represents 56% of the whole dataset. Unlike the NewsEye data, the EMM corpus does not provide the position of the target entities. Additionally, each stance specifies the opinion expressed on the whole related text towards a given target. Table 5 shows sample EMM corpus annotations.

News ID	Quotation	Source Name	Source ID	Target ID	Target Name	Ann1	Ann2	Agreement
1	It's time for Gordon Brown and his colleagues to break the spell Peter ...	Peter Kilfoyle	16361	36	Gordon Brown			1
2	I view reversal of the Bush Administration's 2001 policy as an essential ...	David Gold	59395	1	George W. Bush	NEG	NEG	1
3	We remind people that there was, indeed, good news under President Bush, ...	Fleischer	247210	1	George W. Bush	POS	POS	1

Table 5: Examples of EMM corpus annotations.

The PULS corpus contains 17,354 different documents extracted from business news articles with 19,689 company names, among them 14,172 are distinct instances. However, unlike NewsEye data which defines three classes of stances, the PULS corpus uses 5 polarities to categorised stances: 1.0 for very positive, 0.7 somehow positive, 0.5 for neutral, 0.3 somehow negative and 0.0 for very negative. We, therefore, adapt this corpus to our work by merging the classes somehow positive and very positive to belong to the same class (positive). As for the 'negative' class, it includes very negative and somehow negative classes.

Different information is provided in the PULS corpus. A list of of related named entities along with their detailed properties are associated to each article; For each named entity, these properties include 'entityId', 'name', positions 'offsets' and 'polarity'. An example of this corpus is shown in Figure 3. Although the PULS dataset contains information about the start/end position of named entities inside each textual content, stances towards chosen targets are expressed over the text as a whole. In other words, even if the same target named entity is mentioned several times, only one stance will cover all of its mentions. In this work, we ignored stances from the EMM and the PULS corpora having a disagreement between annotators.


```
[ {"content": "German prosecutors widen market... "
   "docnoId": "2C39ABDDB47F3B7C32E9D668C6186EF2",
   "entities": [{"entityId": 5094,
                 "name": "Volkswagen",
                 "offsets": [{"end": 106, "start": 96},
                             {"end": 162, "start": 152}],
                 "polarity": 0.0},
                {"entityId": 10458,
                 "name": "Audi",
                 "offsets": [{"end": 1630, "start": 1626}],
                 "polarity": 0.0}],
   "headline": {"end": "125", "start": "42"}
  },
]
```


Figure 3: Samples of PULS corpus.

Table 6 summarises the external datasets used in this work. It shows the number of stances defined and their distribution according to their classes.

	POS	NEG	NEUT	Contradiction	Total
EMM-dataset	193	234	866	299	1,592
PULS-dataset	10,127	9,214	377	146	19,864

Table 6: Distribution of stances according to their classes in the external datasets.

All the datasets used in this work are divided into three subsets (∼80% for training, ∼10% for development and ∼10% for testing). Because, unlike EMM and PULS, the NewsEye data set contains a much larger number of neutral stances than positive and negative ones, we removed 50% of the neutral stance annotations from the training data. We expect that this reduces the impact of the imbalanced data problem, and has a positive impact on our performance at detecting positive and negative stance. As explained in Section 5.3, this is particularly important since for posterior applications, it is more useful to detect those than to detect neutral stances. The development set is used in order to tune the parameters of the supervised learning approaches.

5 Experimental results

The quality of each method is measured by precision, recall and F1-score to evaluate the results for each stance class (positive, negative or neutral). Precision P is the rate of stances correctly classified by the system. Recall R is the rate of stances present in the reference that is found and correctly classified by the system. Finally, the F1-score is the harmonic mean between precision and recall:

$$F1 = \frac{2 * P * R}{P + R} \tag{4}$$

In addition, we used the accuracy and the F1-macro to know how each system performs overall across the datasets. The accuracy indicates the rate of the total number of correct predicted stances compared to the total number of stances while the F1-macro is the average of the per-class F1-scores.

5.1 Lexicon-based approach

This section reports on the results of our lexicon-based approach, which computes the polarity score of named entities relying on the polarity and number of surrounding opinion words (cf. Section 3.1).

Our lexicon-based methods are designed to determine the stance towards given positions of named entities. The named entity position is very important information because we exploit neighbouring words. However, the position of named entities is unavailable in the EMM corpus, where the target may not even be mentioned in the text. Therefore, the EMM dataset cannot be used to evaluate the lexicon-based approach, and it is ignored in this section. Furthermore, if the dataset only provides a global stance towards a target entity, like the English PULS dataset, our method is slightly modified. In this case, the score of the target entity is computed as the average of the scores of each of its mentions.

The number of neighbour words (n) and the threshold (k) are chosen based on the experiments with the development data of each dataset. For the NewsEye datasets, they are set to $n = 1$ and $k = 0.1$

for fi-NewsEye, fr-NewsEye and sv-NewsEye, and to $n = 2$ and $k = 0.1$ for de-NewsEye. For the PULS dataset, their values are set to $n = 17$, $k = 0.07$.

Table 7 provides the detailed results obtained with our lexicon-based method. Its performance highly depends on the importance of the class. Indeed, the stance class with the highest number of instances gets the highest precision, recall, and F1-score.

		P (%)	R (%)	F1 (%)	Support
de-NewsEye	POS	04.08	20.00	06.78	10
	NEG	06.98	17.65	10.00	17
	NEUT	97.48	90.09	93.64	858
	Accuracy (%)		87.91		885
	F1-macro (%)		36.81		885
fi-NewsEye	POS	0	0	0	5
	NEG	16.67	25.00	20.00	4
	NEUT	96.81	94.19	95.48	258
	Accuracy (%)		91.39		267
	F1-macro (%)		38.49		267
fr-NewsEye	POS	0	0	0	4
	NEG	0	0	0	2
	NEUT	99.53	82.46	90.19	1,026
	Accuracy (%)		81.98		1,032
	F1-macro (%)		30.06		1,032
sv-NewsEye	POS	0	0	0	5
	NEG	0	0	0	1
	NEUT	97.26	89.50	93.22	238
	Accuracy (%)		87.30		244
	F1-macro (%)		31.07		244
PULS-dataset	POS	97.52	65.45	78.33	2,521
	NEG	24.09	70.37	35.89	216
	NEUT	00.23	25.00	00.47	4
	Accuracy (%)		65.78		2,741
	F1-macro (%)		38.23		2,741

Table 7: Performance of the lexicon-based approach. “Support” stands for the number of occurrences of each stance class in the test set.

5.2 Machine learning approach

Machine learning approaches including feature-based machine learning (cf. Section 3.2.1) and deep learning methods (cf. Section 3.2.2) are evaluated across all the datasets: the NewsEye NE dataset v1, the EMM dataset and the PULS dataset.

5.2.1 Feature-based ML approach

The accuracy of the stance classification task using feature-based classifiers are shown in Table 8. The classifiers are more accurate on the NewsEye dataset. However, the F1-macro averages (cf. Table 9) are lower than those calculated on the other datasets. This is not unexpected with the imbalance between stance classes. Classifiers typically tend to predict the dominant class. Results show that the linear discriminant analysis classifier (LDA) outperforms other classifiers in all the NewsEye datasets as well as the external datasets. The logistic regression classifier optimised by the stochastic gradient

	de-NewsEye	fi-NewsEye	fr-NewsEye	sv-NewsEye	EMM dataset	PULS dataset
LR	96.95	96.63	99.42	98.35	60.81	59.64
SGD	97.18	96.63	99.42	98.35	57.89	60.83*
K Nearest	95.93	93.63	99.42	98.35	52.64	53.19
SVM	96.95	96.63	99.42	98.35	58.48	60.20
RF	96.95	95.51	96.37	98.35	51.51	48.05
Adaboost	96.95	96.63	95.45	98.35	56.67	55.44
DT	94.92	95.51	92.54	98.35	59.11	42.77
LDA	97.40*	**97.75***	99.42	98.35	61.44*	60.83*
Bayes	96.05	95.13	96.90	98.35	40.37	41.56
Majority vote	**97.74**	96.63	**99.42**	**98.35**	**61.64**	**63.12**

Table 8: Accuracy of feature-based ML classifiers

descent (SGD) has the highest accuracy in the PULS dataset. However, the table shows that all the classifiers have comparable results regardless of the dataset, with the exception of the decision tree that generates results with considerable difference between the EMM and the PULS corpora. On the (small) NewsEye Swedish dataset, the stances predicted by all the classifiers are neutral. Furthermore, except for the fi-NewsEye dataset, we obtained better accuracy using the majority vote classifier than using any one of the classifiers separately. The performance of the majority vote classifier are detailed in Table 9.

		P	R	F1	Support
de-NewsEye	POS	50.00	30.00	37.50	10
	NEG	40.00	23.53	29.63	17
	NEUT	97.90	98.95	98.42	858
	Accuracy (%)	97.74			885
	F1-macro (%)	55.18			885
fi-NewsEye	POS	0	0	0	5
	NEG	0	0	0	4
	NEUT	96.63	100	98.29	258
	Accuracy (%)	96.63			267
	F1-macro (%)	32.54			267
fr-NewsEye	POS	0	0	0	4
	NEG	0	0	0	2
	NEUT	99.42	100	99.71	1,026
	Accuracy	99.42			1,032
	F1-macro	33.24			1,032
sv-NewsEye	POS	0	0	0	5
	NEG	0	0	0	1
	NEUT	98.35	100	99.17	238
	Accuracy (%)	98.35			244
	F1-macro (%)	33.06			244
EMM-dataset	POS	50.90	46.67	48.69	60
	NEG	42.86	31.03	36.00	29
	NEUT	68.53	75.38	67.98	130
	Accuracy (%)	61.64			130
	F1-macro (%)	50.89			219
PULS-dataset	POS	64.80	65.93	65.36	2,521
	NEG	38.64	31.48	31.48	216
	NEUT	0	0	0	4
	Accuracy (%)	63.12			2,741
	F1-macro (%)	32.28			2,741

Table 9: Performance of the majority vote classifier.

Table 9 shows that the majority vote classifier reaches good results on the NewsEye datasets. However, it is not able to predict positive and negative stances when they are trained on too few such subjective stances. Similarly, the majority of the feature-based ML classifiers fail to detect the four neutral opinions in the PULS dataset. When the distribution of stances according to their classes is balanced (i.e. the EMM corpus), the classifier gives fairly good results.

5.2.2 Deep learning approach

We assess our deep learning method on the same data. Our models are trained with the optimizer AdamW, learning rate as 5e-5, and a higher number of epochs than recommended (10). We tuned batch size and maximum sequence length based on the development data. Batch size is set to 16 for EMM and PULS datasets, and to 32 for NewsEye datasets.Maximum sequence length is assigned to 256 for EMM and PULS data, and to 128 for NewsEye data. The other hyperparameters of BERT models are unchanged.

		Uncased BERT model			Cased BERT model			
		P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	Support
de-NewsEye	POS	25.00	20.00	22.22	27.78	50.00	35.71	10
	NEG	08.33	05.88	06.90	14.29	11.76	12.90	17
	NEUT	97.23	98.02	97.62	97.77	97.20	97.49	858
	Accuracy (%)		**95.37**			95.03		885
	F1-macro (%)		42.25			**48.70**		885
fi-NewsEye	POS	100.0	20.00	33.33	100.0	80.00	88.89	5
	NEG	0	0	0	0	0	0	4
	NEUT	96.99	100.0	98.47	98.10	100.0	99.04	258
	Accuracy (%)		97.00			**98.13**		267
	F1-macro (%)		43.94			**62.64**		267
fr-NewsEye	POS	0	0	0	0	0	0	4
	NEG	0	0	0	0	0	0	2
	NEUT	99.41	100.0	99.71	99.41	100.0	99.71	1,026
	Accuracy (%)		99.42			99.42		1,032
	F1-macro (%)		33.24			33.24		1,032
sv-NewsEye	POS	71.42	100.0	83.33	35.71	100.0	52.63	5
	NEG	50.00	100.0	66.67	0	0	0	1
	NEUT	100.0	98.74	99.37	100.0	96.64	98.29	238
	Accuracy (%)		**98.77**			96.31		244
	F1-macro (%)		**83.12**			50.31		244
EMM-dataset	POS	62.16	38.33	47.42	52.17	20.00	28.92	60
	NEG	36.17	58.62	44.74	40.00	20.69	27.27	29
	NEUT	69.63	72.31	70.94	62.98	87.69	73.31	130
	Accuracy (%)		**61.19**			60.27		219
	F1-macro (%)		**54.37**			43.17		219
PULS-dataset	POS	97.35	93.10	95.17	97.76	88.42	92.86	2,521
	NEG	58.14	69.44	63.29	44.27	76.85	56.18	216
	NEUT	01.39	25.00	02.63	01.16	25.00	02.22	4
	Accuracy (%)		**91.13**			87.41		2,741
	F1-macro (%)		**53.70**			50.41		2,741

Table 10: Performance of deep learning approach.

Table 10 shows the results of this model with detailed evaluation for each type of stance. Our approaches based on BERT models have the same difficulties as the feature-based ML ones in processing

the NewsEye dataset, because it contains few subjective stances in comparison with a large number of neutral stances. This is particularly true for fr-NewsEye, which has the highest relative share of stances annotated to neutral in the NewsEye NE dataset v1, and for which the deep learning approach did not propose any positive or negative stance. Likewise, no negative stance was found for fi-NewsEye, while at the same time a very good score is obtained for positive stances (F-measure 89%). The performance of our method is better when dealing with the EMM and PULS datasets which have a more balanced distributions across stance classes.

Comparing the cased and uncased BERT models, the uncased model globally surpassed the cased one. Indeed, the uncased model obtains a better F1-macro on 3 datasets (sv-NewsEye, EMM and PULS) in English and Swedish, and comparable scores on fr-NewsEye. However, the cased BERT model overperformed the uncased one on the German and Finnish NewsEye datasets. We believe this may have to do with the larger share of cased words in those languages, but a larger volume of data would be needed to accurately conclude about the respective performances of the uncased and cased BERT models in detecting stance over multilingual datasets.

5.3 Discussion

Table 11 summarises the results of all our methods. Machine learning methods clearly surpass the lexicon-based method on all the datasets. Based on F1-macro average score, the feature-based ML method outperforms both of the deep learning models on German, whereas on English, Finnish and Swedish the best scores are given by the BERT models. On French data, all machine learning methods assigned a neutral stance for all the named entities, which lead to identical scores for all of them.

Dataset	Lexicon-based		Feature-based ML		Deep learning			
					Uncased BERT model		Cased BERT model	
	Accuracy	F1-macro	Accuracy	F1-macro	Accuracy	F1-macro	Accuracy	F1-macro
de-NewsEye	87.91	36.81	**97.74**	**55.18**	95.37	42.25	95.03	48.70
fi-NewsEye	91.39	38.49	96.63	32.54	97.00	43.94	**98.13**	**62.64**
fr-NewsEye	81.98	30.06	**99.42**	**33.24**	**99.42**	**33.24**	**99.42**	**33.24**
sv-NewsEye	87.30	31.07	98.35	33.06	**98.77**	**83.12**	96.31	50.31
EMM-dataset	–	–	**61.64**	50.89	61.19	**54.37**	60.27	43.17
PULS-dataset	65.78	38.23	63.12	32.28	**91.13**	**53.70**	87.41	50.41

Table 11: Summary of performances of all experimented approaches, highlighting the top-performing score of each data set in bold font

Although it is difficult to decide from these results which method is more suitable to proceed with the automatic stance annotation of the whole NewsEye corpus, we think that BERT models are more appropriate than feature-based ML approaches. Indeed, as shown by the detailed results (Tables 9 and 10), the BERT models are better at predicting less-seen stance classes on the training corpus (i.e. positive and negative stances) while the feature-based model usually tends to assign the dominant stances on which it is trained (i.e. neutral stance). Considering the NewsEye data only, we recommend using BERT models, in particular the cased BERT model which reaches better results on the datasets in Finnish and German. We will however monitor potential changes as we obtain additional training data. An extended version of the NewsEye NE dataset (v2) is indeed expected by the summer 2020.

6 Conclusion

In this deliverable, we have conducted and evaluated a series of experiments on stance classification with different ways of representing the target named entity and the body text. Two approaches have been tested. First, we developed a lexicon-based approach which uses a list of negative/positive words. This approach does not require a training dataset and showed a good accuracy especially on smaller and imbalanced datasets such as the NewsEye NE dataset v1. Stance towards named entities is evaluated by relying on neighbouring words. We used a window of $2 \times n$ words where the n words preceding and following the target respectively define its left and right contexts. The polarities of these words allowed extracting the stance towards the target named entity. This approach showed reasonable results on the NewsEye datasets (more than 80% accuracy, around 35% F1-macro on average), which are characterised by the dominance of the neutral class.

The second approach is language-independent, relying on supervised machine learning. Two methods have been tested, one of them based on a majority vote classifier among 9 feature-based machine learning models and the other based on a robust-to-noise deep learning BERT model. These methods relied on text classification where the body text and the target entity were converted into vectors and then classified into stance classes. The results confirmed the relevance of the majority vote classifier as the best feature-based ML approach. Over the NewsEye dataset, the results of the BERT models and the majority vote classier are close, with better performance for BERT on the Swedish and Finnish subsets, better performance for the majority vote classifier in German, and comparable scores for French. The main difference, however, is that the BERT models perform better over the negative and positive stance classes.

The final goal of Task T3.2 is to determine the best tool to extract the stance towards named entities over the collections of historical newspapers provided by NewsEye partner libraries. Among the different methods tested in this work, the cased BERT model always showed good results on all the NewsEye datasets as well as the external datasets used. In addition, benefits of this model are that it is language-independent, robust to noise, and does not rely on handcrafted rules. Therefore, we intend to use the cased BERT model to perform stance detection within the NewsEye workflow. It will be applied over the whole NewsEye collection, and the subsequent automatic stance annotation will be made available through the NewsEye demonstrator. This way, the stance annotations will be accessible both directly to users, and through APIs to subsequent software tools, such as the ones developed within WP4 and WP5. The code of all the tools presented in this deliverable is available on Github [6], while research publications and datasets may be found in the NewsEye community of Zenodo [7].

The present deliverable is the final version of Task T3.2 on stance detection, however the work holds some promising prospects. Once stances will be computed over the whole NewsEye data, it will be interesting to visualise the evolution of the stance over various parameters such as time, media and country. Another potential future work consists in using our tools to detect the stance towards other targets, pre-defined by DH scholars or other users. Some terms and concepts are known to be polarising and shall particularly help them to answer opinion-related research questions.

[6] https://github.com/NewsEye/Stance-Detection
[7] https://zenodo.org/communities/newseye/

References

[1] Kazi Saidul Hasan and Vincent Ng. "Stance classification of ideological debates: Data, models, features, and constraints". In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 2013, pp. 1348–1356.

[2] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. "Cats rule and dogs drool!: Classifying stance in online debate". In: *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*. Association for Computational Linguistics. 2011, pp. 1–9.

[3] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. "Semeval-2016 task 6: Detecting stance in tweets". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016, pp. 31–41.

[4] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. "Discourse-aware rumour stance classification in social media using sequential classifiers". In: *Information Processing & Management* 54.2 (2018), pp. 273–290.

[5] Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. "A dataset for multi-target stance detection". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. 2017, pp. 551–557.

[6] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In: *Lrec*. Vol. 10. 2010. 2010, pp. 2200–2204.

[7] GuangXu Zhou, Hemant Joshi, and Coskun Bayrak. "Topic Categorization for Relevancy and Opinion Detection." In: *TREC*. 2007.

[8] Claire Gautsch and Jacques Savoy. *UniNE at TREC 2008: Fact and opinion retrieval in the blogsphere*. Tech. rep. Neuchatel Univ (Switzerland), 2008.

[9] Kerstin Denecke. "Using sentiwordnet for multilingual sentiment analysis". In: *2008 IEEE 24th international conference on data engineering workshop*. IEEE. 2008, pp. 507–512.

[10] Yanqing Chen and Steven Skiena. "Building sentiment lexicons for all major languages". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2014, pp. 383–389.

[11] Henrik Bøhler, Petter Asla, Erwin Marsi, and Rune Sætre. "IDI @ NTNU at SemEval-2016 Task 6: Detecting Stance in Tweets Using Shallow Features and GloVe Vectors for Word Representation". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016, pp. 445–450.

[12] Heba Elfardy and Mona Diab. "Cu-gwu perspective at semeval-2016 task 6: Ideological stance detection in informal text". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016, pp. 434–439.

[13] Martin Tutek, Ivan Sekulić, Paula Gombar, Ivan Paljak, Filip Čulinović, Filip Boltužić, Mladen Karan, Domagoj Alagić, and Jan Šnajder. "Takelab at semeval-2016 task 6: Stance classification in tweets using a genetic algorithm based ensemble". In: *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*. 2016, pp. 464–468.

[14] Sara S Mourad, Doaa M Shawky, Hatem A Fayed, and Ashraf H Badawi. "Stance detection in tweets using a majority vote classifier". In: *International Conference on Advanced Machine Learning Technologies and Applications*. Springer. 2018, pp. 375–384.

[15] Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. “pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016, pp. 384–388.

[16] Yuki Igarashi, Hiroya Komatsu, Sosuke Kobayashi, Naoaki Okazaki, and Kentaro Inui. “Tohoku at SemEval-2016 task 6: feature-based model versus convolutional neural network for stance detection”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016, pp. 401–407.

[17] Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluıés Màrquez, and Alessandro Moschitti. “Automatic stance detection using end-to-end memory networks”. In: *arXiv preprint arXiv: 1804.07581* (2018).

[18] Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. “Stance detection with bidirectional conditional encoding”. In: *arXiv preprint arXiv:1606.05464* (2016).

[19] Guido Zarrella and Amy Marsh. “Mitre at semeval-2016 task 6: Transfer learning for stance detection”. In: *arXiv preprint arXiv:1606.03784* (2016).

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[21] Lidia Pivovarova, Arto Klami, and Roman Yangarber. “Benchmarks and models for entity-oriented polarity detection”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. 2018, pp. 129–136.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[23] Valeriya Slovikovskaya. “Transfer Learning from Transformers to Fake News Challenge Stance Detection (FNC-1) Task”. In: *arXiv preprint arXiv:1910.14353* (2019).

[24] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. “Learning Word Vectors for 157 Languages”. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[25] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. “Enriching word vectors with subword information”. In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[27] Stefan Schweter and Johannes Baiter. “Towards robust named entity recognition for historic german”. In: *arXiv preprint arXiv:1906.07592* (2019).

[28] George-Alexandru Vlad, Mircea-Adrian Tanase, Cristian Onose, and Dumitru-Clementin Cercel. “Sentence-Level Propaganda Detection in News Articles with Transfer Learning and BERT-BiLSTM-Capsule Model”. In: *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. 2019, pp. 148–154.

[29] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. “Google’s neural machine translation system: Bridging the gap between human and machine translation”. In: *arXiv preprint arXiv:1609.08144* (2016).

[30] Paula Carvalho, Luıés Sarmento, Jorge Teixeira, and Mário J Silva. “Liars and saviors in a sentiment annotated corpus of comments to political debates”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics. 2011, pp. 564–568.

[31] Swapna Somasundaran, Josef Ruppenhofer, and Janyce Wiebe. “Discourse level opinion relations: An annotation study”. In: *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. 2008, pp. 129–137.

[32] Nolan Lawson, Kevin Eustice, Mike Perkowitz, and Meliha Yetisgen-Yildiz. “Annotating Large Email Datasets for Named Entity Recognition with Mechanical Turk”. In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*. Los Angeles: Association for Computational Linguistics, 2010, pp. 71–79.

[33] Jacob Cohen. “A coefficient of agreement for nominal scales”. In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[34] Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. “Sentiment analysis in the news”. In: *arXiv preprint arXiv:1309.6202* (2013).