xLiMe
FP7-ICT-2013-10
ICT-2013.4.1
Contract no.: 611346
www.xlime.eu

Deliverable D2.3.2

Final Text from Social Media Prototype

Editor:	Luis Rei, JSI
Author(s):	Luis Rei, JSI
Deliverable Nature:	Prototype (P)
Dissemination Level:	Public (PU)
Contractual Delivery Date:	M24 – 31 October 2014
Actual Delivery Date:	M24 – 31 October 2014
Suggested Readers:	All project partners
Version:	1.0
Keywords:	Text from social media; final prototype; Natural Language Processing; Text mining

Disclaimer

Full Project Title:	xLiMe – crossLingual crossMedia knowledge extraction
Short Project Title:	xLiMe
Number and Title of Work package:	WP2 Text Extraction from Multilingual Multimedia Natural Language
Document Title:	D2.3.2 - Final Text from Social Media Prototype
Editor:	Luis Rei, JSI
Work package Leader:	Blaž Novak, JSI

Executive Summary

The deliverable presents the final prototype for processing text from social media streams. The main objectives of this deliverable are to produce a representation that allows further statistical processing including integration with information extracted from other sources such as video and knowledge bases. For a given social media message, the developed prototype generates as its output both a vector-space-model representation of the text, a normalized version of the text and annotations: Part of Speech (PoS) tags, Named Entities (NE) and sentiment polarity.

The year 1 prototype had focused exclusively on English language messages. The new prototype provides full support for 3 languages: English, German, Spanish and partial support for Italian. In this deliverable, we present the corpus we created to evaluate the new annotation software and its results.

The corpus, scripts and software have all been available as open source to other researchers and companies.

Table of Contents

Table of Figures

Table of Tables

Abbreviations

NE	Named Entity
NER	Named Entity Recognition
PoS	Part of Speech
SVM	Support Vector Machine
SGD	Stochastic Gradient Descent
PV	Paragraph Vector
PVDM	Paragraph Vector Directed Memory

1 Introduction

This deliverable describes the methods implemented, criteria for the choice of technology and evaluations performed related to the extraction of information from social media text, as well as the prototype developed to perform it in the scope of the xLiMe project. It continues the work done described in D2.3.1 Early Text From Social Media Prototype.

1.1 Background and Motivation

Social media has made following developing news stories anywhere in the world both easy and popular. Tens, sometimes hundreds, of millions of social network users can view photos, videos and comments from people at the scenes of an event almost instantly. Publishers and readers alike post links to news articles about the events to the social networks, providing users with more information at the distance of a click. In the realm of entertainment, TV shows, movies premiering in theatres, concerts, live shows and conferences are now accompanied by the live comments of their viewers. Brands and their marketing departments have widely acknowledged both the positive and negative potential of social networks.

There are three use cases in xLiMe that will provide feedback end evaluation for the system to be developed. They will be run by the following partners:

- **ZATTOO** is the biggest Internet (OTT) TV provider in Europe. Headquartered in Zurich, Switzerland, the company is present in six European countries, with a clear focus on Switzerland and Germany. Zattoo intends to enhance its existing products with information and automatic cross-media recommendation technology provided by xLiMe.

- **VICO Research & Consulting GmbH** is a social media monitoring provider with multinational companies as clients. VICO's platform focus on providing fast access to text generated online, mainly from social networks, in multiple languages, and importantly to the analysis performed on them. VICO plans to improve its product with cross-media and cross-language technology developed by xLiMe.

- **ECONDA GmbH** provides web-analytics and recommendation solutions specialized to E-commerce businesses. Geographically its core market are the DACH countries: Germany, Austria and Switzerland. Currently ECONDA covers more than 30% of the German E-Commerce market. ECONDA hopes to improve the range of solutions it offers to its customers with technologies developed in xLiMe.

The use cases selected in xLiMe are all relevant to this deliverable and prototype. Respectively:

- **Cross-media content enrichment**: Providing multimedia content consumers with additional related content enhances the service provided by ZATTOO. Among the additional related content is text based content from social media detailed in this deliverable, speech detailed in Deliverable D2.1.2 and video detailed in Deliverable 2.2.2.
- **Cross-media analytics**: The information and annotations extracted from social media text by this deliverable will enhance analytics with named entities and sentiment and improve collection of related social media documents as well as help match with other types of data such as video channels.
- **Cross-media product recommendations**: This deliverable will help enhance the recommendations provided by E-commerce sites by extracting relevant data from social media which will be integrated with the data extracted from other sources such as main stream news articles and TV channels.

For more details on the use case partners, their businesses and plans for technologies developed in xLiMe see xLiMe Deliverables D1.4.2 Requirements for Demonstrator and D8.2.2 Draft Business Plan.

1.2 Challenges

While the need to real-time monitor and integrate social media streams into news, entertainment and companies' public relations and marketing efforts is widely acknowledged, several technical challenges complicate such integration. The first and the most obvious is volume: with 500M posts per day on the Twitter network alone, processing the social media stream efficiently in close to real-time is non-trivial. The second challenge is inherent to the social media text itself: a single short piece of text, which usually relies on external context and is often lacking the syntactical correctness and regularity easily found in mainstream media articles written and edited by teams of professionals. Finally, social media is a truly international phenomenon with more than 35 languages represented on twitter alone. While ideally any solution developed in xLiMe would work with any language, the use cases make certain languages more relevant than others, namely, German, English, Spanish and Italian were selected to be supported by this prototype.

2 Text from Social Media

For the year 2 prototype that expanded the text processing to multiple languages, we have developed a new piece of software, twitter annotator[1], which handles the routing of a given language to its respective processing pipeline as well as the distribution of work among processes – either in a single computer or multiple computers if necessary.

Further the prototype's functionality is divided into separate components. Each component is independent from the others and can be used as a separate tool. The components are:

- The Load Balancing Router and the Pipeline Workers
- The Tokenizer and Pre-processor
- The sentiment classifier (a general text classifier)
- Sequence Tagging (PoS/NER) and Vector Representation

2.1 Requirements

2.1.1 Functional Requirements

The functional requirements of the Final Social Media from Text prototype are derived from the use case scenarios and the project's Description of Work.

- **Normalized text** is expected to help with cross-media matching;
- **Vector Representation** to help with cross-media and cross-lingual matching by allowing further statistical processing;
- **Message level sentiment polarity** is important for both recommendations and social media monitoring;
- **Named Entity Recognition (NER)** is important for cross-media and cross-lingual matching, for social media monitoring, information enrichment and providing better recommendations;
- **Part of Speech Tagging** is expected to provide a valuable feature for further information extraction;
- **Multiple Language Support** - German and English language text is essential, support for other languages is desirable. Given the use case partners, the additional language we chose for immediate support was Spanish. Partial support Italian also exists. As we had planned in the Future Work chapter of Deliverable D2.3.1 we investigated adding coverage for a smaller language, namely Slovenian. However, the lack of use cases supportive of the endeavour and the additional time required to build a corpus, models and evaluation made this infeasible in the context of this prototype.

2.1.2 Quality Requirements

Beyond functional requirements for this prototype there are also highly desirable quality requirements:

- **Deployment**: the prototype should be easy to deploy with minimal external dependencies so that it can easily be integrated into our infrastructure and easy to move to our partner's infrastructure and easy to adopt by third parties.
- **Maintainability**: it should be easy to isolate and correct issues with the prototype. Where external dependencies exist, these should be in active development or extremely stable.

[1] https://github.com/lrei/twitter_annotator

- **Interoperability**: the prototype should integrate easily with the existing project infrastructure and, secondarily, any other infrastructure in use by JSI, the partners and third parties.
- **Extensibility**: the prototype should be architecture in such a way that makes it relatively easy to add support for other languages and other domains of text as well as new features in the form of new annotations or the extraction of additional information from the input.
- **Reliability & Availability**: the year 2 prototype should be reliable. In stream processing, results in gaps of service and information analysis that are deeply undesirable in the year 2 prototype.
- **Efficiency**: this prototype is expected to be able to handle high volume of data in close to real-time.
- **Scalability**: the amount of data processed by this prototype is expected to increase with the increase use of social media. Our use case partners are also expected to grow their businesses. As such, the prototype should be able to make good use of better hardware (vertical scaling) as well as be able to scale horizontally.
- **Open Source**: it is the philosophy of the xLiMe project to make the software and resources developed Open Source to allow other researchers to use the technologies we develop. It is also our philosophy to adopt other Open Source software where possible.

The requirement of ease of deployment, maintainability and extensibility where important in selecting the Python language to write the prototype in as code executes in a cross-platform interpreter and has its own dependency management solutions that are easy to use. The language is also very widely used, including inside JSI and the partners. It was designed specifically to be easy to use. Extensibility, Interoperability and Scalability where significant factors in the design of the software architecture of the prototype.

2.2 Architecture Overview

The model routing is achieved via a configuration file while the process routing is done using zeroMQ[2]. The flow diagram is shown in Figure 1: a request to process a social media message arrives at the router and is directed to a process, local or remote, which processes the message and replies to the request. The fact that more workers can be added which are on different computers helps achieve some amount of horizontal scalability.

[2] http://zeromq.org/

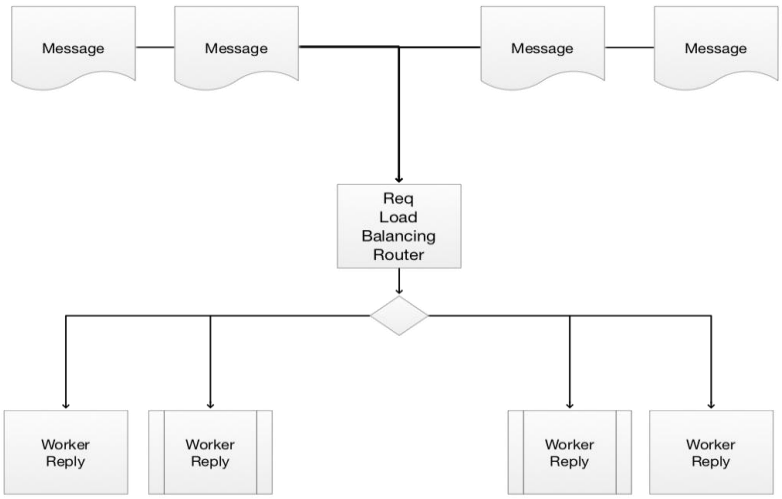

Figure 1: Twitter Annotator flow diagram

The worker processes the message by routing it to its language specific pipeline similar to what was done in the year 1 prototype and described in Deliverable D2.3.1. The flow within the pipeline for a language is shown in Figure 2. The pipeline is built from a configuration file and the worker's code. In the next sections we will detail each part of this pipeline.

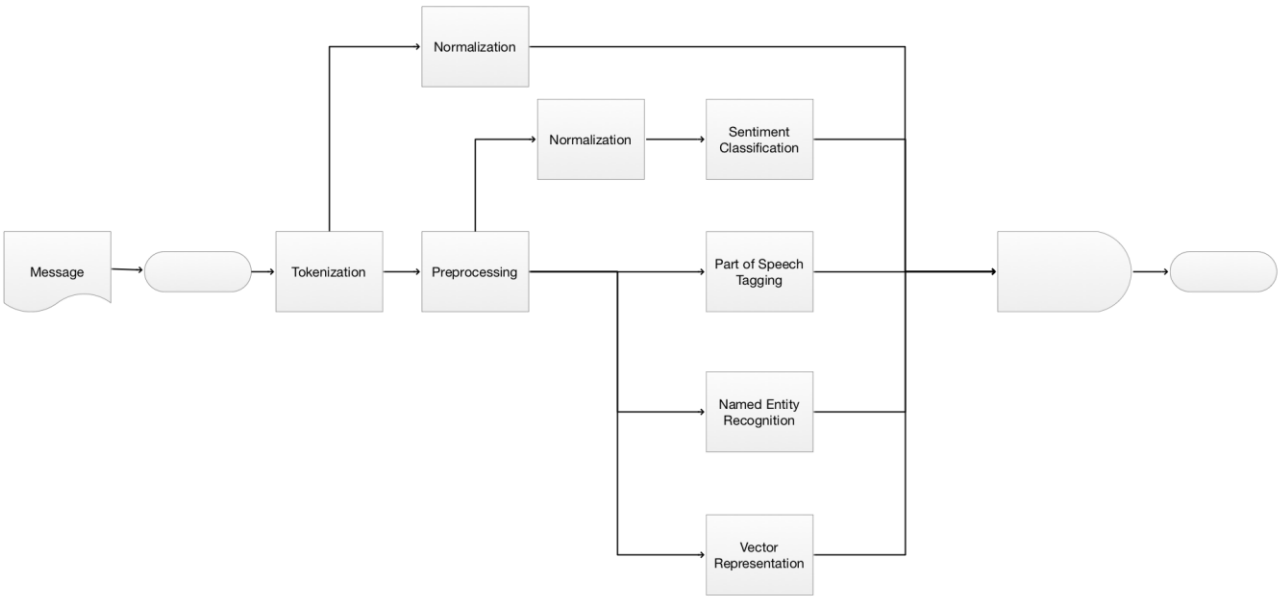

Figure 2: Language Processing Pipeline

2.3 Tokenization

Tokenization consists of splitting a document into tokens, mostly individual words. This is often slightly harder on social media than news text due to common conventions not being followed, the presence of strings of punctuation used in unconventional ways (e.g. smileys, +====) and the presence of Unicode glyphs (e.g. U+2620, skull and crossbones). These can make standard tokenizers work poorly. There are additional differences due to twitter's use of in-band signalling such as "@-mentions" and "#hashtags" and the pervasive presence of URLs.

Following year 1's prototype, we opted for using a version Tweetmotif's [1] tokenizer, commonly called twokenizer, that was modified by CMU Ark[3] and renamed ark-twokenize, ported back to python by Myle Ott[4] and refered to as ark-twokenize-py, and subsequently slightly modified for xLiMe including the option to segment tokens with apostrophes differently for both compatibility with English language resources and Italian linguistic practice (e.g. "l'ammore" becomes "l'" "ammore").

2.4 Pre-processing

The light Pre-processing we perform consists of lower-casing text and replacing some patterns, corresponding to usernames (@-mentions), numbers and URLs with predefined tokens. We also split the hash "#" in hahshtags from the rest of the token. The pre-processor is also responsible for fixing HTML escaping errors that are commonly found in tweets such as replacing "&" with "&".

2.5 Normalization

Normalization hand stop-word removal as not changed much from the year 1 prototype: text is case-folded, the text is Unicode normalized and stop-words and punctuation characters are removed. We also use the normalized text to generate N-grams (unigrams, bigrams and trigrams).

2.6 Sentiment Classification

The social media sentiment classifier used in this prototype was developed in the Symphony project[5] and is detailed in the Symphony Deliverable D2.2 [2]. It is a message level sentiment polarity classifier that for each message outputs one of three labels: negative, neutral (/objective) or positive.

In order to be able to support as many languages as possible, the corpus used to train the classifier is automatically generated using the following approach:

1. Tweets are collected randomly;
2. Tweets whose language property does not match the desired language are discarded;
3. Tweets are tokenized, pre-processed and normalized;
4. Optionally, tweets that have too few tokens, too many URLs or @-mentions are discarded;
5. Optionally langid.py[6] is used to further filter according to desired language;
6. Tweets with only positive emoticons are stored and labelled as positive examples, tweets with only negative emoticons are stored and labelled as negative examples, and all others are discarded;

[3] http://www.ark.cs.cmu.edu/TweetNLP/
[4] https://github.com/myleott/ark-twokenize-py
[5] http://projectsymphony.eu/
[6] https://github.com/saffsd/langid.py

7. The emoticons in a percentage of the tweets are removed;
8. Tweets from accounts of mainstream media sources for the language are collected, tokenized and pre-processed, labelled neutral and stored.

The software used for generating the corpus is also Open Source[7].

Using this automatically generated corpus, a classifier for each language was trained using unigram, bigram and trigram features. The classifier is a Stochastic Gradient Descent (SGD) trained linear Support Vector Machine (SVM) implemented using scikit-learn[8] it is a very fast algorithm.

2.7 Named Entity Recognition

Named Entity Recognition consists of identifying tokens in a document which are names in a predefined category: names of people, organizations and locations. There is commonly a catch-all category for other names to be recognized, usually named "Miscellaneous". For example:

$[\text{Steve Jobs}]_{\text{PERSON}}$ unveiled $[\text{Apple}]_{\text{ORGANIZATION}}$'s $[\text{iPhone}]_{\text{MISCELLANEOUS}}$ in $[\text{San Francisco}]_{\text{LOCATION}}$.

We use the Stanford NER tagger[9] from Python via the NLTK[10] wrapper. The Stanford NLP open source software is widely used in industry, academia, and government. It's often the standard against which other tools are often compared. It supports several languages including German, English and Spanish.

As described in the Future Work chapter of xLiMe Deliverable D3.2.1, we also built a Neural Network classifier based on the window level classifier described in [3] that uses only word vectors as inputs. The word vectors used as input were trained using the CBOW [4] algorithm and both Wikipedia and Twitter text as a corpus.

2.8 Part of Speech Tagging

Parts of Speech tagging consists of classifying tokens in a document according to their part of speech category such as noun, verb, adjective, etc. We use the Stanford POS tagger[11] from Python via the NLTK wrapper. It supports several languages including German, English and Spanish.

2.9 Vector Representation

The vector representation is intended to allow further matching cross-media matching and allow for statistical processing algorithms to use it. In the year 1 prototype we had implemented a Bag of N-Grams together with the hashing trick to generate a sparse vector. In the Future Work chapter of Deliverable D2.3.1 we mentioned we wanted to implement a different approach, namely Paragraph Vectors [5] which we have for year 2. The implemented model is the Paragraph Vector Directed Memory (PVDM) model

[7] https://github.com/lrei/twitter_sentiment_gen
[8] http://scikit-learn.org
[9] http://nlp.stanford.edu/software/CRF-NER.shtml
[10] http://www.nltk.org/
[11] http://nlp.stanford.edu/software/tagger.shtml

trained on a corpus that mixes Wikipedia and tweets for the language in question. The PVDM model provides an ordered low rank representation of an arbitrary length of text.

3 Evaluation

We have evaluated the multiple sub-systems of our year 2 prototype according to standard metrics for each sub-component using standard corpora, when available, for each task. Because for Twitter text there aren't many linguistic resources available as exist for mainstream news, we also built a human annotated corpus for German, Spanish and Italian with sentiment polarity, part of speech and named entity tags.

For sentiment we used the Semeval 2014 Task 9[12] metrics: precision, recall and their F1 calculated ignoring correctly classified neutral/objective documents. For NER we use the CoNLL 2003 metrics: precision, recall and their F1 calculated using exact matching and ignoring correctly classified "outside" tokens. We used the CoNLL 2000[13] script to perform the evaluations. Finally, for PoS we use per token accuracy measured on all tokens.

3.1 Corpora

3.1.1 Existing Corpora

To evaluate the sentiment classifier on English tweets we used a subset of the Semeval 2014 Task 9 corpus. The corpus is publicly distributed as a list of tweet ids and their manually annotated sentiment labels negative, neutral/objective, positive. Because tweets can be made unavailable (either by the twitter user or by twitter itself) after they were annotated and their ids and labels distributed, there is no guarantee that someone else can retrieve all of the tweets at a later date. The subset we used was retrieved on August of 2015. It was subsequently under-sampled to be balanced at 445 tweets per label.

For evaluating NER on English tweets we used the Ritter Twitter corpus [6]. Because we use 4 classes (Person, Organization, Location and Miscellaneous) for NER we had to convert the 10-class annotations used in the Ritter corpus to the 4 classes according to the methodology described in [7].

For evaluating PoS on English tweets we also used the Ritter Twitter corpus. For PoS we use universal PoS tags[14] extended to Twitter data following similar scheme to the TweeBank [8].

3.1.2 xLiMe Twitter Corpus

The xLiMe Twitter Corpus[15] is a manually annotated corpus built specifically to evaluate the year 2 prototype for German, Spanish and Italian. The corpus covers all three evaluated tasks: sentiment polarity, NER and PoS. The document and token counts of the corpus are shown in Table 1.

For evaluating sentiment, we under-sampled the corpus to contain the same number of examples for each class in each language.

Language	Number of Annotated Tweets	Number of Annotated Tokens
German	3,447	58,264
Spanish	10,000	174,151
Italian	86,46	154,371

Table 1: xLiMe Twitter corpus document and token counts

[12] http://alt.qcri.org/semeval2014/task9/
[13] http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt
[14] http://universaldependencies.github.io/docs/u/pos/index.html
[15] https://github.com/lrei/xlime_twitter_corpus

3.2 Criteria and Selected Subsystems

When selecting candidates for the different components of the system, both functional and quality requirements were used as criteria. Support for the project's main languages, English and German, was the first priority while support for secondary and other languages was less important.

A second priority was creating a system that was reliable and easy to maintain. There had been reliability issues with the year 1 prototype. Towards this end, the principle of "keep it simple" was guided our development of the year 2 prototype. The entire system should be small and avoid unnecessary complexity. This led us to avoid any sub-component that was not written in Python or wrapped from Python and had to instead rely on other methods of integration such as web services.

The sentiment polarity classifier we developed in Symphony was the only open source sentiment classifier that supported all the languages we needed out of the box while providing an easy path to add support for more. It was also easy to integrated and use. We opted for Stanford NLP tools based on the support for the languages we wanted and the ease of integration with the rest of the infrastructure as well as their reputation for quality – since we started evaluating their usage we did not having a single issue.

3.3 Results

Table 2 shows the results of the evaluation of our sentiment classifier while Table 3 shows the results

Language	F1 (semeval 2014)	Examples
English	66%	445 per class
German	61%	142 per class
Spanish	63%	195 per class
Italian	54%	521 per class

Table 2: Sentiment Polarity Classifier Results: Semeval 2014 corpus for English, xLiMe Twitter corpus for other languages

Model	Language	Precision	Recall	F1	Speed
twitter_nlp *reported results on 3 classes averaged 4 cross validation folds	English	73%	49%	59%	NA
Stanford NER (conll.distsim.iob2.crf.ser.gz)	English	35%	34%	35%	3k tokens/s
NN-WLL **averaged 4 cross validation folds	English	21%	31%	26%	500k tokens/s
Stanford NER (german.hgc_175m_600.crf.ser.gz)	German	34%	35%	34%	11k tokens/s
Stanford NER (spanish.ancora.distsim.s512.crf.ser.gz)*** evaluated without tag IB prefix	Spanish	22%	43%	28%	16k tokens/s

Table 3: NER Results Ritter Twitter corpus for English, xLiMe Twitter corpus for other languages

4 Conclusion

In year 2 we implemented support for multiple languages. We created a new twitter corpus containing PoS, NER and sentiment labels for German, Spanish and Italian and addressed all that points of the Future Work chapter of D2.3.1.

We developed software for capable of all the annotations required (sentiment, PoS, NER, PVs, normalization), routing each peace of text to the appropriate pipeline according to language and capable doing load balancing to distribute the work load among multiple computers enabling easy horizontal scaling. We also improved the maintainability, extendibility, reliability and ease of deployment over the previous year prototype and released this prototype as open source.

We implemented an extremely fast alternative PoS/NER tagger that can be easily trained to new domains and new languages at the expense of performance (precision/recall) but that can be used if the need for a much faster Pos/NER tagging arises.

We evaluated our prototype using a combination of existing corpora and the corpus we built. While English-only performance is down from the year 1 prototype, that is a trade-off we were willing to make in order to have support for multiple languages without adding adding too much complexity and in order to have a more reliable prototype.

All the models, software, corpora, scripts and evaluation results can be found at http://xlime.ijs.si/t23/.

References

[1] Brendan, Michel Krieger, and David Ahn O'Connor, "TweetMotif: Exploratory Search and Topic Summarization for Twitter.," in *ICWSM*, 2010.

[2] Symphony Project, Symphony Deliverable D2.2.

[3] R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P Collobert, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2403-2537, 2010.

[4] Kai Chen, Greg Corrado, and Jeffrey Dean Tomas Mikolov, "Efficient Estimation of Word Representations in Vector Space," in *ICLR*, 2013.

[5] Quoc, and Tomas Mikolov Le, "Distributed Representations of Sentences and Documents," in *ICML*, 2014.

[6] Alan and Clark, Sam and Mausam and Etzioni, Oren Ritter, "Named Entity Recognition in Tweets: An Experimental Study," in *EMNLP*, 2011.

[7] Diana Maynarda, Giuseppe Rizzob, d, Marieke van Erpc, Genevieve Gorrella, Raphaël Troncyb, Johann Petraka, Kalina Bontchevaa Leon Derczynskia, "Analysis of named entity recognition and linking for tweets," *Information Processing & Management*, pp. 32–49, 2015.

[8] Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider and Noah A. Smith. Olutobi Owoputi, "Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters," in *NAACL*, 2013.

