Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text*

Alexandre Pinto[1], Hugo Gonçalo Oliveira[2], and Ana Oliveira Alves[3]

1 CISUC, Dept. of Informatics Engineering, University of Coimbra, Coimbra, Portugal
arpinto@student.dei.uc.pt

2 CISUC, Dept. of Informatics Engineering, University of Coimbra, Coimbra, Portugal
hroliv@dei.uc.pt

3 CISUC, Dept. of Informatics Engineering, University of Coimbra, Coimbra, Portugal; and
Polythecnic Institute of Coimbra, Coimbra, Portugal
aalves@isec.pt, ana@dei.uc.pt

Abstract

Nowadays, there are many toolkits available for performing common natural language processing tasks, which enable the development of more powerful applications without having to start from scratch. In fact, for English, there is no need to develop tools such as tokenizers, part-of-speech (POS) taggers, chunkers or named entity recognizers (NER). The current challenge is to select which one to use, out of the range of available tools. This choice may depend on several aspects, including the kind and source of text, where the level, formal or informal, may influence the performance of such tools. In this paper, we assess a range of natural language processing toolkits with their default configuration, while performing a set of standard tasks (e.g. tokenization, POS tagging, chunking and NER), in popular datasets that cover newspaper and social network text. The obtained results are analyzed and, while we could not decide on a single toolkit, this exercise was very helpful to narrow our choice.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases Natural language processing, toolkits, formal text, social media, benchmark

Digital Object Identifier 10.4230/OASIcs.SLATE.2016.3

1 Introduction

The Web is a large source of data, mostly expressed in natural language text. Natural language processing (NLP) systems need to understand the human languages in order to extract new knowledge and perform diverse tasks, such as information retrieval, machine translation, or text classification, among others. For widely-spoken languages, such as English, there is currently a wide range of NLP toolkits available for performing lower-level NLP tasks, including tokenization, part-of-speech (POS) tagging, chunking or named entity recognition (NER). This enables that more complex applications do not have to be developed

* This work was supported by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) – within project REMINDS – UTAP-ICDT/EEI-CTP/0022/2014.

5th Symposium on Languages, Applications and Technologies (SLATE'16).
Editors: Marjan Mernik, José Paulo Leal, and Hugo Gonçalo Oliveira; Article No. 3; pp. 3:1–3:16
Open Access Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

completely from scratch. Yet, with the availability of many such toolkits, the one to use is rarely obvious. Users have also to select the most suitable set of tools that meets their specific purpose. Among other aspects, the selection may consider the community of users, frequency of new versions and updates, support, portability, cost of integration, programming language, the number of covered tasks, and, of course, their performance. During the previous process of selection, the authors of this paper ended up comparing a wide range of tools, in different tasks and kinds of text. This paper reports the comparison of well-known NLP toolkits and their performance in four common NLP tasks – tokenization, POS tagging, chunking and NER – in two different kinds of text – newspaper text, typically more formal, and social network text, often less formal. Although the majority of the tested tools could be trained with specific corpora and / or for a specific purpose, we focused on comparing the performance of their default configuration, which means that we used the available pre-trained models for each tool and target task. This situation is especially common for users that either do not have experience, time or available data for training the tools for a specific purpose. Besides helping us to support our decision, we believe that this comparison will be helpful for other developers and researchers in need of making a similar selection.

The remainder of this paper starts with a brief reference on previous work. After that, the tasks where the toolkits were compared are enumerated, which is followed by the description of the datasets used as benchmarks, all of them previously used in other evaluations. The measures used for comparison are then presented, right before its results are reported and discussed. Although there was not a toolkit that outperformed the others in all the tested tasks and kinds of text, this analysis revealed to be very useful, as it narrowed the range of possible choices and lead to our current selection.

2 Related Work

In academic, official or business contexts, written documents typically use formal language. This means that syntactic rules and linguistic conventions are strictly followed. On the other hand, although typically used orally, informal language has become frequent in written short messages or posts in social networks, such as Facebook or Twitter. In opposition to news websites, where posts are more elaborated, complex and with a higher degree of correctness, in text posted in social networks, it is common to find shorter and simpler sentences that tend to break some linguistic conventions (e.g. proper nouns are not always capitalized, or punctuation is not used properly), make an intensive use of abbreviations, and where slang and spelling mistakes are common. For instance, in informal English, it is common to use colloquial expressions (e.g. “look blue”, “go bananas”, “funny wagon”), contractions (e.g. “ain’t”, “gonna”, “wanna”, “y’all”), clichés (e.g. “An oldie, but a goodie”, “And they all lived happily ever after”), slang (e.g. “gobsmacked”, “knackered”), abbreviations (e.g. “lol”, “rofl”, “ty”, “afaik”, “asap”, “diy”, “rsvp”); the first and the second person, imperative (e.g. “Do it!”) and usually active voices, in addition to the third person and the passive voice, which are generally the only in formal text. Informal language poses an additional challenge for NLP tools, most of which developed with formal text on mind and significantly dependent on the quality of the written text. Given the huge amounts of data transmitted everyday in social networks, the challenge of processing messages written in informal language has received much attention in the later years. In fact, similarly to well-known NLP shared tasks based on corpora written in formal language, including the CoNLL-2000, 2002 or 2003 shared evaluation tasks[25] , tasks using informal text have also been organized, including, for instance, the Making Sense of Microposts Workshop (MSM 2013)[1] or tasks included in the SemEval workshops (e.g. Sentiment Analysis from Twitter [23]).

González [13] highlights the particular characteristics of Twitter messages that make common NLP tasks challenging, such as irregular grammatical structure, language variants and styles, out-of-vocabulary words or onomatopeias, reminding the fact that there is still a lack of gold standards regarding colloquial texts, especially for less-resourced languages.

Besides comparing different NLP tools, in this work, we also analyze their performance in different types of text, some more formal, from newspapers, and some less formal, from Twitter. Other comparisons have been made by others, including the following. In order to combine different NER tools and improve recall, Dlugolinský et al. [7] assessed selected tools for this task in the dataset of the MSM2013 task. This included the comparison of well-known tools such as ANNIE[2], OpenNLP[3], Illinois Named Entity Tagger[4] and Wikifier[5], OpenCalais[6], Stanford Named Entity Tagger[7] and Wikipedia Miner[8].

Godin et al. [12] also used the MSM2013 challenge corpus and performed similar evaluations oriented to NER web services, such as AlchemyAPI[9], DBpedia Spotlight[10], OpenCalais, and Zemanta[11]. Since the evaluated services use complex ontologies, a mapping between the obtained ontologies and entity types was performed, with good F_1 scores when using AlchemyAPI for the person (78%) and location (74%) type entities, and OpenCalais for the organization (55%) and miscellaneous (31%) entities. Rizzo et al. [20] also evaluated web services, such as Lupedia[12], Saplo[13], Wikimeta[14] and Yahoo Content Analysis (YCa), but with focus on different kinds of well-formed content and varying length, such as TED talks transcripts, New York Times articles and abstracts from research papers. In fact, they evaluated the resulting NER and Disambiguation (NERD) framework, which unified the output results of the aforementioned web services, supporting the fact that tools such as AlchemyAPI, OpenCalais and additionally DBpedia Spotlight perform well in well-formed contents, using formal language. Rizzo et al. also report on the evaluation of datasets with colloquial text, namely Twitter text from the MSM2013 challenge and newspaper text from the CoNLL-2003 Reuter Corpus [21]. They report better NER results when using a combination of the tested tools, achieving F_1 results greater than 80% on he CoNLL-2003 dataset, for all entity types and F_1 results greater than 50% on the MSM-2013 dataset, except for the miscellaneous type that obtained results less than 30%.

Garcia and Gamallo [10] report the development of a multilingual NLP pipeline. To assess the performance of the presented tool, they performed experiments with POS-tagging and NER. The POS-tagger performed slightly better than well-known tools such as OpenNLP and Stanford NER, achieving a precision score of 94% on the Brown Corpus. On the other

[1] http://microposts2016.seas.upenn.edu
[2] https://gate.ac.uk/sale/tao/splitch6.html#chap:annie
[3] https://opennlp.apache.org
[4] https://cogcomp.cs.illinois.edu/page/software_view/NETagger
[5] https://cogcomp.cs.illinois.edu/page/software_view/Wikifier
[6] http://www.opencalais.com
[7] http://nlp.stanford.edu/software/CRF-NER.shtml
[8] http://wikipedia-miner.cms.waikato.ac.nz
[9] http://www.alchemyapi.com
[10] https://github.com/dbpedia-spotlight/dbpedia-spotlight
[11] http://www.zemanta.com
[12] http://dbpedia.org/projects/lupedia-enrichment-service
[13] http://saplo.com
[14] https://www.w3.org/2001/sw/wiki/Wikimeta

hand, the NER module achieved F_1 scores of 76% and 59% on the IEER[15] and SemCor[16] Corpus, respectively.

Rodriquez et al. [22] and Atdag and Labatut [1] compared different NER tools applied to different kinds of text, respectively biographical and OCR texts. Rodriquez et al. used Stanford CoreNLP, Illinois NER, LingPipe and OpenCalais, on a set of Wikipedia biographic articles annotated with person, location, organization and date type entities. Due to the absence of biography datasets, the evaluated corpus was fully designed by the authors, i.e, the evaluated corpus consisted of a series of Wikipedia articles which were annotated with the aforementioned entity types. Although CoreNLP obtained the best F_1 scores (60% and 44%) in two manually-annotated resources, there was not a tool that outperformed all the others in every entity type. They are rather complementary. Atdag and Labatut evaluated OpenNLP, Stanford CoreNLP, AlchemyAPI and OpenCalais using datasets with the entity types person, location and organization manually annotated. They used data from the Wiener Library, London and King's College London's Serving Soldier archive, which consisted of Holocaust survivor testimonies and newsletters written for the crew of H.M.S. Kelly in 1939. Once again, Stanford CoreNLP gave the best overall F_1 results (90%) while OpenCalais only achieved 73%.

3 Addressed Tasks

In order to evaluate how good standard NLP tools perform against different kinds of text, such as noisy text from social networks and formal text from newspapers, we performed a set of experiments where the performance in common NLP tasks was analysed. The addressed tasks were tokenization, POS-tagging, chunking and NER. The following list describes the four evaluated tasks:

- *Tokenization:* usually the first step in NLP pipelines. It is the process of breaking down sentences into tokens, which can be words or punctuation marks. Although it seems a relatively easy task, it has some issues because some words may rise doubts on how they should be tokenized, namely words with apostrophes, or with mixed symbols.
- *Part-of-Speech (POS) Tagging:* given a specific tagset, it determines the part-of-speech of each token in a sentence. In this work, the tags of the Penn Treebank Project [17], popular among the NLP community, are used.
- *Chunking*: also known as shallow parsing, it is a lighter syntactic parsing task. The main purpose is to identify the constituent groups in which the words are organized. This includes at least noun phrases (NP), verb phrases (VP) and prepositional phrases (PP). The sequence of chunks forms the entire sentence. They may also be nested inside each other to form a tree structure, where each leaf is a word, the previous node is the corresponding POS-tag and the head of the tree is the chunk type.
- *Name Entity Recognition/Classification*: deals with the identification of certain types of entities in a text and may go further classifying them into one of given categories, typically PERson, LOCations, ORGanizations, all proper nouns, and sometimes others, such as dates.

These are common NLP tasks, the first step of several more complex NLP applications, and supported by several NLP toolkits for English, including those compared in this work.

[15] http://www.itl.nist.gov/iad/894.01/tests/ie-er/er_99/er_99.htm
[16] http://www.gabormelli.com/RKB/SemCor_Corpus

Listing 1 Example of the Annotated Data Format.

```
Token        POS      Syntactic Chunk    Named Entity
Only          RB        B-NP               O
France        NNP       I-NP               LOC
and           CC        I-NP               O
Britain       NNP       I-NP               LOC
backed        VBD       B-VP               O
Fischler      NNP       B-NP               PER
's            POS       B-NP               O
proposal      NN        I-NP               O
.             .         O                  O
```

4 Used Datasets

In order to evaluate the performance of the different NLP toolkits and determine the best performing ones, the same criteria must be followed, including the same metrics and manually-annotated gold standard data. Testing tools in the same tasks and scenarios makes comparison fair and more reliable. For this purpose, we relied on well-known datasets widely used in NLP and text classification research, not only in the evaluation of NLP tools, but also for training new models. More precisely, we used different gold standard datasets that cover different kinds of text – newspaper and social media. Regarding newspaper text, we used a collection of news wire articles from the Reuters Corpus[17], previously used in the shared task of the 2003 edition of the CoNLL conference. The POS and chunking annotations of this dataset were obtained using a memory-based MBT tagger [5]. The named entities were manually annotated at the University of Antwerp [25].

In order to represent social and more informal text, we first used the annotated data from Alan Ritter's Twitter corpus[18], with manually tokenized, POS-tagged and chunked Twitter posts, also with annotated named entities. The collection of Twitter posts used in the MSM 2013 workshop[19], where named entities are annotated, was also used as a gold standard for social media text.

The POS tags of the CoNLL-2003 dataset follow the Penn Treebank style [20]. Alan Ritter's corpus follows the same format, with the same POS-tags and additional specific tags for retweets, @usernames, #hashtags, and urls. For the chunk tags, the format I|O|B-TYPE is used in both datasets. This is interpreted as: the token is inside (I), in the beginning (B) of a following chunk of the same type or outside (O) of a chunk phrase [18]. The named entities in the CoNLL-2003 dataset are annotated using four entity types, namely Location (LOC), Miscellaneous (MISC), Organization (ORG) and Person (PER). In Alan Ritter's corpus, entity types were not exactly the same, so they had to be converted, as we mention further on this section. The #MSM2013 corpus only contains annotated named entities and their types. To ease experimentation, this corpus was converted to the same format as the other two.

Listing 1 illustrates the annotation format for the experiments. Table 1 shows some numerical characteristics of the used datasets.

[17] http://trec.nist.gov/data/reuters/reuters.html
[18] https://github.com/aritter/twitter_nlp/tree/master/data/annotated
[19] http://oak.dcs.shef.ac.uk/msm2013/challenge.html
[20] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Table 1 Dataset properties.

Dataset	*Documents*	*Tokens*	*Average Tokens per Document*
CoNLL (Reuter Corpus)	946	203621	215
Twitter (Alan Ritter)	2394	46469	19
#MSM2013	2815	52124	19

Table 2 Datasets with PoS Tags.

	Dataset				
	Twitter (Alan Ritter)		CoNLL (Reuter Corpus)		Description
CC	305	(2.01 %)	3653	(1.79 %)	Coordinating conjunction
CD	268	(1.76 %)	19704	(9.68 %)	Cardinal number
DT	825	(5.43 %)	13453	(6.61 %)	Determiner
IN	1091	(7.18 %)	19064	(9.36 %)	Preposition or subordinating conjunction
JJ	670	(4.41 %)	11831	(5.81 %)	Adjective
MD	181	(1.19 %)	1199	(0.59 %)	Modal
NN	1931	(12.72 %)	23899	(11.74 %)	Noun, singular or mass
NNP	1159	(7.63 %)	34392	(16.89 %)	Proper noun, singular
NNS	393	(2.59 %)	9903	(4.86 %)	Noun, plural
PRP	1106	(7.28 %)	3163	(1.55 %)	Personal pronoun
PRP$	234	(1.54 %)	1520	(0.75 %)	Possessive pronoun
RB	680	(4.48 %)	3975	(1.95 %)	Adverb
RT	152	(1.00 %)	0		Retweet
TO	264	(1.74 %)	3469	(1.70 %)	to
UH	493	(3.25 %)	30	(0.01 %)	Interjection
URL	183	(1.21 %)	0		Url
USR	464	(3.06 %)	0		User
VB	660	(4.35 %)	4252	(2.09 %)	Verb, base form
VBD	306	(2.02 %)	8293	(4.07 %)	Verb, past tense
VBG	303	(2.00 %)	2585	(1.27 %)	Verb, gerund or present participle
VBN	140	(0.92 %)	4105	(2.02 %)	Verb, past participle
VBP	527	(3.47 %)	1436	(0.71 %)	Verb, non-3rd person singular present
VBZ	342	(2.25 %)	2426	(1.19 %)	Verb, 3rd person singular present
Others	908	(5.98%)	10478	(5.15 %)	

It is clear that the Twitter datasets (Alan Ritter and #MSM2013) have a greater number of documents with short sentences. On the other hand, the CoNLL dataset has longer and more complex sentences. Tables 2 and 3 show the distribution of the POS and chunk tags, respectively for Alan Ritter's and CoNLL-2003 corpora. For the POS tags, only those that account for more than one percent at least in one of the two datasets, excluding punctuation marks, are shown. Noun phrases (NP), prepositional phrases (PP) and verbal phrases (VP) are the most common chunks in both datasets.

For the NER evaluation, we stripped the IOB tags from the datasets whenever they were present, and joined them in a single entity tag, i.e, different tags such as B-LOC and I-LOC became simply LOC. Besides making comparison easier, this was made due to some noticed inconsistencies on the usage of I's and B's. Table 4 shows the distribution of the named entities in all of the used datasets.

We recall that the entity types in Alan Ritter's corpus are more and different than the other two. So, in order to enable comparison in the same lines, additional entity types were considered as alternative tags for one of the types covered by the CoNLL-2003 dataset: LOC,

Table 3 Datasets with Chunk Tags.

	Dataset				
	Twitter (Alan Ritter)		CoNLL (Reuter Corpus)		Description
B-ADJP	241	(1.58 %)	2	(0.00 %)	Begins an adjective phrase
B-ADVP	535	(3.52 %)	22	(0.01 %)	Begins an adverb phrase
B-CONJP	2	(0.01 %)	0		Begins a conjunctive phrase
B-INTJ	384	(2.52 %)	0		Begins an interjection
B-NP	3992	(26.24 %)	3777	(1.85 %)	Begins a noun phrase
B-PP	1027	(6.75 %)	254	(0.12 %)	Begins a prepositional phrase
B-PRT	109	(0.72 %)	0		Begins a particle
B-SBAR	103	(0.68 %)	8	(0.00 %)	Begins a subordinating clause
B-VP	1884	(12.39 %)	163	(0.08 %)	Begins a verb phrase
I-ADJP	86	(0.57 %)	1374	(0.67 %)	Is inside an adjective phrase
I-ADVP	66	(0.43 %)	2573	(1.35 %)	Is inside an adverb phrase
I-CONJP	2	(0.01 %)	70	(0.03 %)	Is inside a conjunctive phrase
I-INTJ	124	(0.82 %)	60	(0.03 %)	Is inside an interjection
I-LST	0		36	(0.02 %)	Is inside a list marker
I-NP	2686	(17.66 %)	120255	(59.06 %)	Is inside a noun phrase
I-PP	10	(0.07 %)	18692	(9.18 %)	Is inside a prepositional phrase
I-PRT	0		527	(0.26 %)	Is inside a particle
I-SBAR	5	(0.03 %)	1280	(0.63 %)	Is inside a subordinating clause
I-VP	842	(5.54 %)	26702	(13.11 %)	Is inside verb phrase
O	27646	(20.47 %)	3113	(13.58 %)	Is outside of any chunk.

MISC, ORG and PER. Table 5 shows the new entities distribution after performing the following mapping: FACILITY, GEO-LOC → LOC; MOVIE, TVSHOW, OTHER → MISC; COMPANY, PRODUCT, SPORTSTEAM → ORG; PERSON, MUSICARTIST → PER. This mapping considered the annotation guidelines of the CoNLL-2003 shared task[21].

5 Compared Tools

In order to select a suitable tool for our purpose, many criteria have to be considered. Among other properties, tools can be implemented in different programming languages; have available models that cover different tasks, kinds of text or languages; require different setups; or have different learning curves for simple usage or for integration. The tools compared in this paper were trained for English and are open, well-known and widely used by the NLP community. Moreover, they were developed either in Java or Python, which, nowadays, are probably the two languages more frequently used to develop NLP applications and for which there is a broader range of available toolkits. The compared tools are enumerated in the following list, where they are described and grouped in "standard" toolkits, which means they were developed with no specific kind of text in mind, and social network-oriented tools, which aim to be used in short messages from social networks.

5.1 Standard NLP toolkits

The *NLTK toolkit*[22] is a Python library aimed at individuals who are entering the NLP field. It is divided in independent modules, responsible for specific NLP tasks such as tokenization,

[21] http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt

[22] http://www.nltk.org

Table 4 Datasets with NER Tags.

	Dataset					
	Twitter (Alan Ritter)		CoNLL (Reuter Corpus)		#MSM2013	
COMPANY	207	(0.45 %)	0		0	
FACILITY	209	(0.45 %)	0		0	
GEO-LOC	325	(0.70 %)	0		0	
LOC	0		8297	(4.07%)	795	(1.53 %)
MISC	0		4593	(2.26%)	511	(0.98 %)
MOVIE	80	(0.17 %)	0		0	
MUSICARTIST	116	(0.25 %)	0		0	
ORG	0		10025	(4.92 %)	842	(1.62 %)
OTHER	545	(1.39 %)	0		0	
PERSON	664	(1.43 %)	11128	(5.47 %)	2961	(5.68 %)
PRODUCT	177	(0.38 %)	0		0	
SPORTSTEAM	74	(0.16 %)	0		0	
TVSHOW	65	(0.14 %)	0		0	
O	44007	(94.70 %)	169578	(83.28 %)	47015	(90.20 %)

Table 5 Dataset with Joint NER Tags.

	Dataset					
	Twitter (Alan Ritter)		CoNLL (Reuter Corpus)		#MSM2013	
LOC	534	(1.15 %)	8297	(4.07 %)	795	(1.53 %)
MISC	690	(1.48 %)	4593	(2.26 %)	511	(0.98 %)
ORG	458	(0.99 %)	10025	(4.92 %)	842	(1.62 %)
PER	780	(1.68 %)	11128	(5.47 %)	2961	(5.68 %)
O	44007	(94.70 %)	169578	(83.28 %)	47015	(90.20 %)

stemming, tree representations, tagging, parsing and visualization. It also comes bundled with popular corpus samples ready to be read. By default, NLTK uses the Penn Treebank Tokenizer, which uses regular expressions to tokenize the text. Its PoS tagger uses the Penn Treebank tagset and is trained on the PENN Treebank corpus with a Maximum Entropy model. The Chunker and the NER modules are trained on the ACE corpus with a Maximum Entropy model [2, 15].

Apache OpenNLP[23] is a Java library that uses machine learning methods for common natural language tasks, such as tokenization, POS tagging, NER, chunking and parsing. Users can either rely on pre-trained models for the previous tasks or train their own with a Perceptron or a Maximum Entropy. The pre-trained models for English PoS tagging and chunking use the Penn Treebank tagset. The Chunker is trained on the CoNLL-2000 dataset. The pre-trained NER models provide cover the recognition of persons, locations, organizations, time, date and percentage expressions. Although there are two POS tagging models available for English, in this work, we used the one based on Maximum Entropy.

The *Stanford CoreNLP*[24] toolkit is a Java pipeline that provides common language processing tasks. The most supported language is English, but other languages are also available [16]. Comparing to other frameworks such as GATE [4] or UIMA [8], CoreNLP is simple to set up and run, since users do not need to learn and understand complex installations and procedures. The CoreNLP performs a Penn Treebank style tokenization and the POS module is an implementation of the Maximum Entropy model using the Penn Treebank tagset. The NER component uses a Conditional Random Field (CRF) model and is trained on the CoNLL-2003 dataset.

[23] `https://opennlp.apache.org`
[24] `http://stanfordnlp.github.io/CoreNLP`

Table 6 Toolkit properties.

System	Programming Language	Target Text	Tokenization	PoS Tagging	Chunking	NER
NLTK	Python	Generic	✓	✓	✓	✓
OpenNLP	Java	Generic	✓	✓	✓	✓
CoreNLP	Java	Generic	✓	✓	✗	✓
Pattern	Python	Generic	✓	✓	✓	✗
TweetNLP	Java	Social Media	✓	✓	✗	✗
TwitterNLP	Python	Social Media	✓	✓	✓	✓
TwitIE	Java	Social Media	✓	✓	✗	✓

Pattern[25] is a Python library that provides modules for web mining, NLP and ML tasks. This library does not provide methods for a single field but rather a general cross-domain and ease-of-use functionality. The PoS tagger uses a simple rule-based model trained on the Brown Corpus [6].

5.2 Social Network-Oriented Toolkits

Alan Ritter's *TwitterNLP*[26] is a Python library that offers a NLP pipeline for performing Tokenization, POS, Chunking and NER. The authors reduced the problem of dealing with noisy texts by developing a system based on a set of features extracted from Twitter-specific POS taggers, a dedicated shallow parsing logic, and the use of gazetteers generated from entities in the Freebase knowledge base, that best match the fleeting nature of informal texts [19].

CMU's *TweetNLP*[27] is Java tool that provides a Tokenizer and a POS Tagger with available models, trained with a CRF model in Twitter data, manually annotated by its authors [11]. In addition to the typical syntactic elements of a sentence, TweetNLP identifies content such as mentions, URLs, and emoticons.

TwitIE[28] is an open-source plugin for GATE. The GATE framework comes already packaged with ANNIE, an information extraction system, and includes resources such as: a Tokenizer, a sentence splitter, gazetteer lists, a PoS tagger and a semantic tagger. TwitIE re-uses some of these components (sentence splitter and gazeteer lists) but adapts the other to the Twitter kind of text, supporting language identification, Tokenization, normalization, PoS tagging and Name Entity Recognition. The TwitIE tokenizer follows the same tokenization scheme as TwitterNLP. The PoS tagger uses an adptation of the Stanford tagger, trained on tweets with the Penn Tree Bank tagset, with additional tags for retweets, URLs, hashtags and user mentions [3]. In our experiments, we used the Text Analytics web service[29] which includes a version of the TwitIE module.

5.3 Tools Summary

Table 6 summarizes additional properties of the aforementioned tools. Java is the most used programming language and only tools such as TweetNLP, TwitterNLP and TwitIE are made

[25] http://www.clips.ua.ac.be/pages/pattern
[26] https://github.com/aritter/twitter_nlp
[27] http://www.cs.cmu.edu/~ark/TweetNLP
[28] https://gate.ac.uk/wiki/twitie.html
[29] http://docs.s4.ontotext.com/display/S4docs/Twitter+IE

with models adapted to the social domain. In terms of task support, NLTK, OpenNLP and TwitterNLP offer a complete NLP pipeline (Tokenization, PoS, Chunking and NER). Without any additional plugin, CoreNLP, TweetNLP and TwitIE lack support for chunking, while Pattern and TweetNLP do not support NER.

6 Comparison Metrics

The performance of a NLP tool in a certain task can be estimated by the quality of its predictions on the classification of unseen data. Predictions made are either considered Positive or Negative (under some category) and expected judgments are called True or False (again, under a certain category). The following are common metrics used to assess classification tasks [24]:

- *Precision:* The proportion of correctly classified instances (True Positives) among all the classified instances under a certain category (True Positives and False Positives).

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

P_i = Precision under Category i
TP_i = True Positives under Category i
FP_i = False Positives under Category i

- *Recall:* The proportion of correctly classified instances (True Positives) under a certain category (True Positives and False Negatives).

$$R_i = \frac{TP_i}{TP_i + FN_i}$$

R_i = Recall under Category i
TP_i = True Positives under Category i
FN_i = False Negatives under Category i

- *F-measure:* Combines precision and recall, and is computed as the harmonic mean between the two metrics.

$$F_1 = \frac{2 \times P_i \times R_i}{P_i + R_i}$$

F_1 = Harmonic Mean
P_i = Precision under Category i
R_i = Recall under Category i

The previous metrics provide insights on the behavior of the tool. We can go further and compute the previous estimations in different ways such as:

- *Micro Averaging:* the entire text is treated as a single document and the individual correct classifications are summed up.

$$P^\mu = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i}$$

P^μ = Micro Precision
C = Set of Classes
TP = True Positives
FP = False Positives

$$R^\mu = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i}$$

R^μ = Micro Recall
C = Set of Classes
TP = True Positives
FP = False Positives

- *Macro Averaging:* the precision and recall metrics are computed for each document and then averaged.

$$P^M = \frac{\sum_{i=1}^{|C|} P_i}{|C|}$$

P^M = Macro Precision
C = Set of Classes

$$R^M = \frac{\sum_{i=1}^{|C|} R_i}{|C|}$$

R^M = Macro Recall
C = Set of Classes

In addition to the previous averages, the standard deviation is a common dispersion metric that may be computed as follows:

$$\sigma = \frac{1}{N-1} \sum_{i=1}^{|N|} (x_i - \bar{x})^2$$

N = Number of samples
x_i = Result of the i-th measurement
$\bar{x}$ = Arithmetic mean of the N results

These evaluation metrics can give different results. Macro averaging weights each class equally, even if there are unbalanced classes. On the other hand, micro averaging weights the documents under evaluation, but it can happen that large classes dominates smaller classes. Therefore, macro averaging provides a sense of effectiveness on small classes, increasing their importance. Of course that selecting the appropriate metric depends on the requirements of the application.

7 Comparison Results

This section reports on the results obtained when performing the addressed tasks on the gold standard datasets, presented earlier, using each toolkit. Tables 7, 8, 9, 10 and 11 show the precision (P), the recall (R) and the F_1-scores for each scenario. The presented results are macro averages, i.e, we computed the precision, recall and F_1 for each document (tweet or news) and then averaged the results. The standard deviations associated with the computed macro-averages (σ) are also presented. Micro averages were not computed because we were more interested in assessing the toolkits performance in different documents and not to use the whole corpus as a large document, which would lower the impact of less frequent tags.

More precisely, each table targets a different task, lines have the results for each tool and there are three columns per corpus (P, R and F_1). Table 7 targets tokenization, Table 8 POS-tagging, and Table 9 chunking. Tables 10 and 11 show two different NER results: entity identification (NER) only considers the delimitation of a named entity, while entity classification (NEC) also considers its given type. Table 11 has an additional line with the results of the best performing system that participated in the CoNLL-2003 shared task [9], which combined four different classifiers (robust linear classifier, maximum entropy, transformation-based learning and a hidden Markov model), resulting in $F_1 = 89\%$ in named entity classification (NEC).

On the CoNLL dataset, which uses formal language, standard toolkits perform well. OpenNLP excels with $F_1 = 99\%$ in tokenization, 88% in POS-tagging and 83% in chunking. In the NER task, NLTK (89%) and OpenNLP (88%) performed closely. TwitterNLP also performed well in this dataset. This is not that surprising if we add that the CoNLL-2003 dataset was one of the corpora TwitterNLP was trained on [19], and it is probably also tuned for this corpus.

As expected, the performance of standard toolkits, developed with formal text in mind, decreases when used in the social network corpora. This difference is between 5-8% for tokenization, 17% for POS-tagging, 17-40% for chunking, or 5-18% for NER. This is not

Table 7 Tokenization Performance Results.

Task	Tokenization					
Data set	CoNLL			Alan Ritter - Twitter		
Metric Tool	P ± σ	R ± σ	F1 ± σ	P ± σ	R ± σ	F1 ± σ
NLTK	0.95 ± 0.11	0.96± 0.10	0.95 ± 0.11	0.83 ± 0.14	0.91 ± 0.09	0.87 ± 0.12
OpenNLP	0.99 ± 0.02	0.99 ± 0.01	0.99 ± 0.02	0.92 ± 0.11	0.96 ± 0.06	0.94 ± 0.08
CoreNLP	0.73 ± 0.31	0.73 ± 0.31	0.73 ± 0.31	0.93 ± 0.13	0.95 ± 0.11	0.94 ± 0.12
Pattern	0.42 ± 0.30	0.41 ± 0.29	0.42 ± 0.29	0.76 ± 0.21	0.78 ± 0.20	0.77 ± 0.20
TweetNLP	0.97± 0.05	0.98 ± 0.02	0.98 ± 0.04	0.96 ± 0.07	0.98 ± 0.05	0.97 ± 0.06
TwitterNLP	0.95 ± 0.10	0.97 ± 0.09	0.96 ± 0.10	0.96 ± 0.07	0.97 ± 0.05	0.96 ± 0.06
TwitIE	0.85 ± 0.15	0.93 ± 0.11	0.89 ± 0.14	0.83 ± 0.16	0.89 ± 0.11	0.86 ± 0.13

Table 8 PoS Performance Results.

Task	PoS Tagging					
Data set	CoNLL			Alan Ritter - Twitter		
Metric Tool	P ± σ	R ± σ	F1 ± σ	P ± σ	R ± σ	F1 ± σ
NLTK	0.65 ± 0.19	0.71 ± 0.18	0.68 ± 0.18	0.65 ± 0.19	0.71 ± 0.18	0.68 ± 0.18
OpenNLP	0.88 ± 0.10	0.88 ± 0.09	0.88 ± 0.10	0.70 ± 0.18	0.73 ± 0.17	0.71 ± 0.17
CoreNLP	0.67 ± 0.29	0.67 ± 0.29	0.67 ± 0.29	0.70 ± 0.19	0.71 ± 0.18	0.71 ± 0.18
Pattern	0.36 ± 0.24	0.35 ± 0.24	0.35 ± 0.24	0.61 ± 0.21	0.62 ± 0.21	0.61 ± 0.20
TweetNLP	0.83 ± 0.10	0.84 ± 0.09	0.84 ± 0.09	0.94 ± 0.08	0.96 ± 0.06	0.95 ± 0.07
TwitterNLP	0.83 ± 0.15	0.84 ± 0.15	0.83 ± 0.15	0.92 ± 0.11	0.93 ± 0.11	0.92 ± 0.11
TwitIE	0.78 ± 0.16	0.85 ± 0.12	0.82 ± 0.14	0.78 ± 0.17	0.84 ± 0.13	0.81 ± 0.14

Table 9 Chunking Performance Results.

Task	Chunking					
Data set	CoNLL			Alan Ritter - Twitter		
Metric Tool	P ± σ	R ± σ	F1 ± σ	P ± σ	R ± σ	F1 ± σ
NLTK	0.70 ± 0.10	0.71 ± 0.10	0.71 ± 0.10	0.51 ± 0.16	0.56 ± 0.16	0.54 ± 0.16
OpenNLP	0.83 ± 0.13	0.83 ± 0.12	0.83 ± 0.12	0.44 ± 0.34	0.46 ± 0.36	0.45 ± 0.39
CoreNLP	n/a	n/a	n/a	n/a	n/a	n/a
Pattern	0.33 ± 0.22	0.32 ± 0.21	0.33 ± 0.21	0.54 ± 0.21	0.56 ± 0.20	0.55 ± 0.20
TweetNLP	n/a	n/a	n/a	n/a	n/a	n/a
TwitterNLP	0.82 ± 0.13	0.84 ± 0.12	0.83 ± 0.13	0.90 ± 0.12	0.91 ± 0.11	0.90 ± 0.11
TwitIE	n/a	n/a	n/a	n/a	n/a	n/a

Table 10 NER Performance Results.

Task	NER					
Data set	CoNLL			Alan Ritter - Twitter		
Metric Tool	P ± σ	R ± σ	F1 ± σ	P ± σ	R ± σ	F1 ± σ
NLTK	0.88 ± 0.12	0.89 ± 0.11	0.89 ± 0.11	0.77 ± 0.16	0.84 ± 0.13	0.80 ± 0.15
OpenNLP	0.88 ± 0.09	0.88 ± 0.08	0.88 ± 0.08	0.85 ± 0.14	0.90 ± 0.11	0.87 ± 0.12
CoreNLP	0.70 ± 0.30	0.70 ± 0.30	0.70 ± 0.30	0.87 ± 0.15	0.89 ± 0.14	0.88 ± 0.15
Pattern	n/a	n/a	n/a	n/a	n/a	n/a
TweetNLP	n/a	n/a	n/a	n/a	n/a	n/a
TwitterNLP	0.88 ± 0.11	0.89 ± 0.10	0.88 ± 0.11	0.96 ± 0.07	0.97 ± 0.05	0.97 ± 0.06
TwitIE	0.74 ± 0.16	0.80 ± 0.14	0.77 ± 0.15	0.77 ± 0.17	0.83 ± 0.14	0.80 ± 0.15

Table 11 NEC Performance Results.

Task	NEC					
Data set	CoNLL			Alan Ritter - Twitter		
Metric Tool	P ± σ	R ± σ	F1 ± σ	P ± σ	R ± σ	F1 ± σ
NLTK	0.84 ± 0.12	0.84 ± 0.12	0.84 ± 0.12	0.75 ± 0.17	0.83 ± 0.14	0.79 ± 0.15
OpenNLP	0.87 ± 0.10	0.87 ± 0.09	0.87 ± 0.09	0.85 ± 0.15	0.89 ± 0.12	0.87 ± 0.13
CoreNLP	0.70 ± 0.30	0.70 ± 0.30	0.70 ± 0.30	0.87 ± 0.16	0.89 ± 0.14	0.88 ± 0.15
Pattern	n/a	n/a	n/a	n/a	n/a	n/a
TweetNLP	n/a	n/a	n/a	n/a	n/a	n/a
TwitterNLP	0.84 ± 0.13	0.85 ± 0.12	0.85 ± 0.12	0.95 ± 0.08	0.96 ± 0.07	0.95 ± 0.08
TwitIE	0.73 ± 0.17	0.80 ± 0.14	0.76 ± 0.16	0.77 ± 0.17	0.84 ± 0.14	0.80 ± 0.15
Florian et al.	0.89	0.89	0.89 ± 0.70	n/a	n/a	n/a

Table 12 NER/NEC Performance Results on the #MSM2013 Data set.

Data set	#MSM2013 - Twitter					
Task	NER			NEC		
Metric Tool	P ± σ	R ± σ	F1 ± σ	P ± σ	R ± σ	F1 ± σ
NLTK	0.83 ± 0.16	0.83 ± 0.16	0.83 ± 0.14	0.85 ± 0.14	0.85 ± 0.15	0.85 ± 0.13
OpenNLP	0.83 ± 0.14	0.86 ± 0.14	0.85 ± 0.14	0.84 ± 0.14	0.86 ± 0.13	0.85 ± 0.13
CoreNLP	0.73 ± 0.19	0.83 ± 0.16	0.78 ± 0.16	0.73 ± 0.19	0.84 ± 0.16	0.78 ± 0.16
Pattern	n/a	n/a	n/a	n/a	n/a	n/a
TweetNLP	n/a	n/a	n/a	n/a	n/a	n/a
TwitterNLP	0.90 ± 0.12	0.90 ± 0.12	0.90 ± 0.12	0.91 ± 0.11	0.91 ± 0.11	0.91 ± 0.11
TwitIE	0.61 ± 0.20	0.73 ± 0.18	0.66 ± 0.18	0.61 ± 0.20	0.73 ± 0.17	0.66 ± 0.18
Habib et al.	0.72	0.80	0.76	0.65	0.73	0.69

the case of Pattern, which performs poorly in the CoNLL corpus but improves significantly when tokenizing, PoS tagging and chunking the Twitter corpora. Although not developed specifically for Twitter, OpeNLP and CoreNLP still obtain interesting results for tokenization and NER in its corpus ($F_1 > 80\%$).

Also as expected, in the Twitter corpus, the Twitter-oriented toolkits performed better than the others. TweetNLP was the best in the tokenization (97%) and POS-tagging (95%) tasks. TwitterNLP performed closely (96% and 92%). In the case of TwitIE, the difference of performance in different types of text was not relevant. Once again, it should be highlighted that TwitterNLP was trained with the Twitter corpus, so this comparison is not completely fair. This is also why we used an additional corpus, #MSM2013, which covers social network text. The results of the NER task in this corpus, shown in table 12, confirm the good performance of TwitterNLP. In the last line of the previous table, we also present the official results of the best system that participated in the #MSM2013 Concept Extraction Task, Habib et al. [14], which apparently underperformed TwitterNLP. Habib et al. combined Conditional Random Fields with Support Vector Machines for recognition and, for classification, each entity was disambiguated and linked to its Wikipedia article, where the category was extracted from.

8 Conclusions

We presented a set of experiments aiming at comparing the performance of different open-domain NLP toolkits, which were used to perform different NLP tasks on different kinds of text, namely news (more formal) and social media text (higher proportion of informal documents).

We have shown that, using only the available pre-trained models, there is not one toolkit that overperformed all the others in every scenario. Though, some are more balanced than the others. Even if it cannot be seen as a strong conclusion, the results suggest that OpenNLP is the best choice for news text, and TwitterNLP for social media text. Although the latter result was biased on the TWitter corpus, where TwitterNLP was trained on, we also tested it on another corpus, where it got the best results. It should be noticed that we ended up using datasets that were more appropriate for specific tasks. For instance, although its text of the CoNLL-2003 dataset is tokenized, POS-tagged, and chunked, it was specifically developed for a NER shared task. On the other hand, we did not use the CoNLL-2000, developed for a chunking shared task. Although this dataset was used to train some of the OpenNLP models, we should also consider its results in the future.

As expected, standard toolkits perform better in formal texts, while Twitter-oriented tools got better results in social media text. Besides helping us to make a selection, we believe that these results might be useful for potential users willing to select the most appropriate tools for their specific purposes, especially if they do not have time or expertise to train new models. Of course, we did not use all the available tools, especially those available as web services, but we tried to cover an acceptable range of widely used toolkits that cover several NLP tasks and developed in two programming languages with a large community – Java and Python. We also regard that, with more available manually annotated datasets, either with formal or informal language, we could always re-train some of the available models and possibly increase the performance achieved with most of the tested tools.

References

1 Samet Atdag and Vincent Labatut. A Comparison of Named Entity Recognition Tools Applied to Biographical Texts. In *Systems and Computer Science (ICSCS), 2013 2nd International Conference on*, pages 228–233, Villeneuve d'Ascq, France, August 2013. IEEE.

2 Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL'06, pages 69–72, Sydney, Australia, 2006.

3 Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A. Greenwood, Diana Maynard, and Niraj Aswani. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics, 2013.

4 Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: An Architecture for Development of Robust HLT Applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 168–175, Philadelphia, Pennsylvania, 2002.

5 Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. MBT: A Memory-Based Part of Speech Tagger-Generator. *arXiv preprint cmp-lg/9607012*, 1996.

6 Tom De Smedt and Walter Daelemans. Pattern for Python. *The Journal of Machine Learning Research*, 13(1):2063–2067, 2012.

7 Štefan Dlugolinský, Peter Krammer, Marek Ciglan, Michal Laclavík, and Ladislav Hluchý. Combining Named Enitity Recognition Tools. In *Making Sense of Microposts (# MSM2013)*, Rio de Janeiro, Brazil, May 2013.

8 David Ferrucci and Adam Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4):327–348, September 2004.

9 Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named Entity Recognition through Classifier Combination. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada, 2003.

10 Marcos Garcia and Pablo Gamallo. Yet Another Suite of Multilingual NLP Tools. In *Languages, Applications and Technologies – Revised Selected Papers of 4th International Symposium SLATE, Madrid, Spain, June 2015*, CCIS, pages 65–75. Springer, 2015.

11 Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers – Volume 2*, pages 42–47, Portland, Oregon, 2011.

12 Fréderic Godin, Pedro Debevere, Erik Mannens, Wesley De Neve, and Rik Van de Walle. Leveraging Existing Tools for Named Entity Recognition in Microposts. In *Making Sense of Microposts (# MSM2013)*, pages 36–39, Rio de Janeiro, Brazil, May 2013.

13 Meritxell González Bermúdez. An analysis of Twitter corpora and the difference between formal and colloquial tweets. In *Proceedings of the Tweet Translation Workshop 2015*, pages 1–7. CEUR-WS. org, 2015.

14 Mena Habib, Maurice Van Keulen, and Zhemin Zhu. Concept extraction challenge: University of Twente at #msm2013. In *Making Sense of Microposts (#MSM2013) Concept Extraction Challenge*, pages 17–20, 2013. URL: `http://ceur-ws.org/Vol-1019/paper_14.pdf`.

15 Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics – Volume 1*, ETMTNLP'02, pages 63–70, Philadelphia, Pennsylvania, 2002.

16 Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, USA, 2014.

17 Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, 19(2):313–330, June 1993. URL: `http://dl.acm.org/citation.cfm?id=972470.972475`.

18 Lance A. Ramshaw and Mitchell P. Marcus. Text Chunking using Transformation-Based Learning. In *Proceedings of the ACL Third Workshop on Very Large Corpora*, pages 82–94, June 1995.

19 Alan Ritter, Sam Clark, and Oren Etzioni. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, July 2011.

20 Giuseppe Rizzo, Raphaël Troncy, Sebastian Hellmann, and Martin Bruemmer. NERD meets NIF: Lifting NLP Extraction Results to the Linked Data Cloud. *LDOW*, 937, 2012.

21 Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web. In *International Conference on Language Resources and Evaluation*, pages 4593–4600, 2014.

22 Kepa Joseba Rodriquez, Mike Bryant, Tobias Blanke, and Magdalena Luszczynska. Comparison of Named Entity Recognition Tools for Raw OCR Text. In *KONVENS*, pages 410–414, Vienna, Austria, 2012.

23 Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, Colorado, June 2015. Association for Computational Linguistics. URL: `http://www.aclweb.org/anthology/S15-2078`.

24 Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, Mars 2002.

25 Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.