Evaluating and Combining Named Entity Recognition Systems

Ridong Jiang
Institute for Infocomm Research, A*STAR
Singapore 138632
rjiang@i2r.a-star.edu.sg

Rafael E. Banchs
Institute for Infocomm Research, A*STAR
Singapore 138632
rembanchs@i2r.a-star.edu.sg

Haizhou Li
Institute for Infocomm Research, A*STAR
Singapore 138632
hli@i2r.a-star.edu.sg

Abstract

Name entity recognition (NER) is an important subtask in natural language processing. Various NER systems have been developed in the last decade. They may target for different domains, employ different methodologies, work on different languages, detect different types of entities, and support different inputs and output formats. These conditions make it difficult for a user to select the right NER tools for a specific task. Motivated by the need of NER tools in our research work, we select several publicly available and well-established NER tools to validate their outputs against both Wikipedia gold standard corpus and a small set of manually annotated documents. All the evaluations show consistent results on the selected tools. Finally, we constructed a hybrid NER tool by combining the best performing tools for the domains of our interest.

1 Introduction

Name entity recognition is an important subtask in natural language processing (NLP). The results of recognition and classification of proper nouns in a text document are widely used in information retrieval, information extraction, machine translation, question answering and automatic summarization (Nadeau and Sekine. 2007; Kaur and Gupta. 2010). Depending on the requirements of specific tasks, the types to be recognized can be person, location, organization and date, which are mostly used in newswire (Tjong et al., 2003), or other commonly used measures (percent, weight, money), email address, etc. It can also be domain specific entity types such as medical drug names, disease symptoms and treatment, etc. (Asma Ben Abacha and Pierre Zweigenbaum, 2001).

Name entity recognition is a challenging task which needs massive prior knowledge sources for better performance (Lev Ratinov, Dan Roth, 2009; Nadeau and Sekine. 2007). Many researches works have been conducted in different domains with various approaches. Early studies focus on heuristic and handcrafted rules. By defining the formation patterns and context over lexical-syntactic features and term constituents, entities are recognized by matching the patterns against the input documents (Rau, Lisa F. 1991; Collins, Michael, Singer, Y. 1999). Rule-based system may achieve high degree of precision. However, the development process is time-consuming and porting these developed rules from one domain to another is a major challenge. Recent research in NER tends to use machine learning approaches (Andrew Borthwick. 1999; McCallum, Andrew and Li, W. 2003; Takeuchi K. and Collier N. 2002). The learning methods include various supervised, semi-supervised and unsupervised learning. The supervised learning tends to be the dominant technique for named entity recognition and classification (David Nadeau and Satoshi Sekine. 2007). However, supervised machine learning methods require large amount of annotated documents for model training and its performance typically depends on the availability of sufficient high quality training data in the domain of interest. There are some systems which use hybrid methods to combine different rule-based and/or machine learning systems for improved performance over individual

Proceedings of the Sixth Named Entity Workshop, joint with 54th ACL, pages 21–27,
Berlin, Germany, August 12, 2016. ©2016 Association for Computational Linguistics

approaches (Srihari R. et al., 2000; Tim R. et al., 2012). Hybrid systems make the best use of the good features of different systems or methods to achieve the best overall performance.

In this paper, we first select several publicly available and well-established NER tools in section 2. Then all the tools are validated in section 3 with CONLL 2003 metrics and a customized partial matching measurement. Then we constructed a hybrid NER system based on the best performed NER tools in section 4.

2 Methodology

2.1 Tool Selection

Our goal is to evaluate freely available NER tools that have good performance for our research projects. The criteria for our selection are as follows:

a) The NER tool is freely available and allows unlimited use.
b) The tool can be downloaded and installed locally and works well with default configuration.
c) The tool is not trained for a specific domain.
d) The tool must be able to recognize the basic three entity types: PERSON, LOCATION, ORGANIZATION

Based on the above criteria, the following NER tools have been selected:

a) Stanford NER (Jenny Rose Finkel et al., 2005).
b) spaCy[1].
c) Alias-i LingPipe (Alias-i. 2008).
d) Natural Language Toolkit (NLTK) (Bird Steven et al. 2009).

2.2 Normalization

The selected tools come with different features, programming languages as well as different tag set and output format. To have an automated and efficient evaluation system, we have to integrate all these tools in one system and normalize all their outputs into a standard format.

Stanford NER is a Java package (version 3.6.0). It is based on linear chain Conditional Random Field (Jenny Rose Finkel et al., 2005). The models were trained on a mixture of CoNLL, MUC-6, MUC-7 and ACE named entity corpora. The basic required output tags are "PERSON", "LOCATION" and "ORGANIZATION".

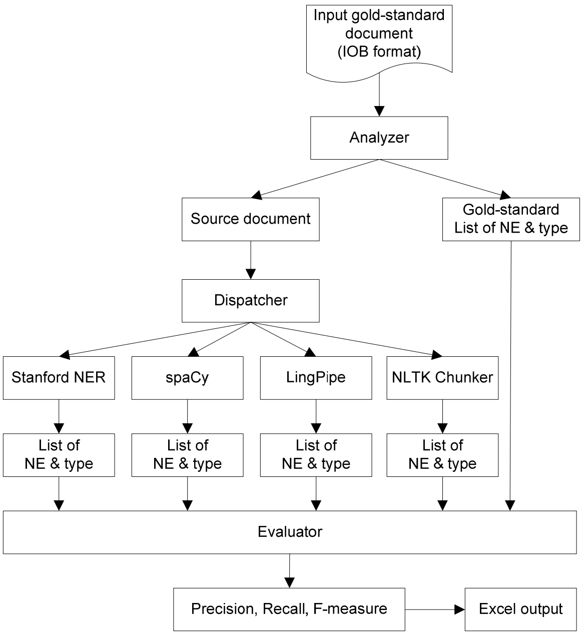
spaCy is implementation in Python. There is no detailed information provided in its documentation with regard to its implemented models at the time of writing. The related output tags include "PERSON", "LOC", "ORG", "GPE" etc. For spaCy outputs, we map "LOC" to "LOCATION", "ORG" and "GPE" to "ORGANIZATION" and ignore all other types.

Alias-i LingPipe NER is implemented in Java and supports both rule-based NER and supervised training of a statistical model or more direct method like dictionary matching (Alias-i. 2008). We use version 4.1.0 and adopt the "First-best Named Entity Chunking". The trained model is News English on the MUC 6 corpus which is relatively slow compared with its other models but with higher accuracy. The output entity types match the normalized types and no mapping is needed.

NLTK is a python NLP toolkit and is well-established in the research community. NLTK's named entity chunker is based on a supervised machine learning algorithm – Maximum Entropy Classifier. Its model is trained on ACE corpus with the exact entity types which we are interested in: "PERSON", "LOCATION" and "ORGANIZATION". It also outputs "GPE" type which we will map to "LOCATION" for evaluation.

2.3 Integration

In order to automate the evaluation process, we developed a system to integrate all the toolkits into one system using a python script.



[1] https://spacy.io/

Fig. 1. System diagram for automated evaluation

The overall system structure of the integrated evaluation system is shown in Fig. 1.

In the process of evaluation, an annotated (gold-standard) input document must be provided. Currently, the supported format is IOB (short for Inside, Outside, Beginning) (Ramshaw and Marcus, 1995). In this scheme, every line in the file represents one token with two fields: the word itself and its named entity type. Empty lines denote sentence boundaries. Following is an example of the representation:

Albert I-PERSON
Einstein I-PERSON
was O
born O
in O
Ulm I-LOCATION
. O

The prefix "I-" in the tag means that the tag is inside a chunk. While the prefix "B-" indicates that the tag is the beginning of a chunk and is only used when a tag is followed by a tag of the same type without "O" tag between them. The "O" tag just means it is out of the chunk. This IOB chunk representation is much easier for manual annotation than inside XML annotation scheme.

An Analyzer module is used to extract the source document as well as all chunks and their types from the annotated file. Every chunk is represented in the format of a three-element tuple: (*chunk, type, start_position*), where the *start_position* is the sequence position (character index) of the chunk in the source document. This tuple representation contains all the necessary information for the validation of a chunk, including its boundary.

The Dispatcher module will pass the source document to all NER tools. All the tools will first tokenize the sentences, analyze these sentences and then create their respective list of tuples dynamically. Every output list from the NER tools will be compared against the standard list generated from the annotated file. The comparison results will be used to calculate true positives (TP), false positives (FP) and false negatives (FN). Then precision, recall and F-measure can be further calculated for evaluation. All the calculation results can be directly exported to excel file for easy comparison.

3 Evaluation

With the methodology defined in section 2, it is ready to evaluate all the selected tools with any data file annotated in the IOB format.

3.1 Evaluation Corpus

Since all the selected NER tools are able to classify the three entity types: PERSON, LOCATION and ORGANIZATION, the evaluation corpus must contain at least the above three entity types. The format is better to be in the supported IOB chunk representation. We found that WikiGold [2] meets the above requirements. WikiGold (Balasuriya et al. 2009) is an annotated corpus over a small sample of Wikipedia articles in CoNLL format (IOB). It contains 145 documents (separated by "-DOCSTART-"), 1696 sentences and 39152 tokens. The statistics of named entities is shown in table 1.

Entity Type	PER	LOC	ORG	MISC	Total
No.	931	1014	898	712	3555

Table 1. WikiGold entities

In the evaluation, we ignore the MISC type and map the gold standard types: PER, LOC and ORG to normalized types PERSON, LOCATION and ORGANIZATION respectively.

3.2 Evaluation Metrics

There are different evaluation metrics for the evaluation of NER systems (Nadeau and Sekine. 2007). The evaluation is basically to check the tool's ability on finding the boundaries of names and their correct types. Most evaluation systems require exact match on both boundary and entity type. The share task for CONLL 2003 (Sang and Meulder, 2003) is one of the examples for the exact matching. However, in some cases, the exact boundary detection is not so important as long as the major part of the name has been identified. For instance, "The United Nations" and "United Nations", "in November 2015" and "November 2015", they are almost the same except the minor differences in the definite article and preposition. The metrics used for evaluation in the Message Understanding

[2] http://downloads.schwa.org/wikiner/wikigold.conll.txt

Conference (MUC) (Grishman and Sundheim, 1996) adopted more loose matching conditions which allow for partial credit when partial span or wrong type detection happened. The credit was given to any correct entity type detected regardless of its boundary as long as there is an overlap, as well as the correct boundary identified regardless of the type. Here we score NER systems based on the following two metrics:

a) Exact matching for both boundary and type (similar to CONLL) which measures a system's capability for accurate named entity detection.
b) Partial matching for boundary is also counted, only when the detected type is correct. This measurement will mitigate the failures of exact matching when the boundary differences are caused by some unimportant words in the names such as the articles and prepositions.

Based on the above two scoring protocols, the measuring system counts TP, FP and FN for every NER toolkit. Then typical precision: p = TP / (TP + FP) and recall: R = TP / (TP + FN) are further calculated to check the NER system's type I (false alarm) and type II (miss) errors respectively.

3.3 Results

		PER	LOC	ORG	OVERALL
Stanford	P	0.7195	0.7753	0.6992	0.7359
	R	0.8733	0.7416	0.4143	0.6813
	F	0.7890	0.7581	0.5203	0.7075
	PP	0.7496	0.8309	0.8083	0.7914
	PR	0.9098	0.7949	0.4788	0.7327
	PF	0.8220	0.8125	0.6014	0.7609
spaCy	P	0.7286	0.7321	0.3346	0.6110
	R	0.7325	0.6144	0.2873	0.5498
	F	0.7305	0.6681	0.3092	0.5788
	PP	0.7788	0.8085	0.5642	0.7240
	PR	0.7830	0.6785	0.4844	0.6514
	PF	0.7809	0.7378	0.5213	0.6858
LingPipe	P	0.4840	0.5067	0.2425	0.4026
	R	0.4211	0.4822	0.2806	0.3985
	F	0.4504	0.4941	0.2602	0.4005
	PP	0.6025	0.6052	0.4341	0.5412
	PR	0.5242	0.5759	0.5022	0.5357
	PF	0.5606	0.5902	0.4657	0.5384
NLTK	P	0.4802	0.4463	0.3115	0.4228
	R	0.7164	0.5493	0.3396	0.5378
	F	0.5750	0.4925	0.3249	0.4734
	PP	0.5587	0.4832	0.4883	0.5136
	PR	0.8335	0.5947	0.5323	0.6532
	PF	0.6690	0.5332	0.5094	0.5750

Table 2. Evaluation results on the WikiGold annotated data for the selected NER tools

Table 2 shows the results of the four selected NER systems on the WikiGold data set.

In the table, Precision (P), Recall (R) and F1 measure (F) are calculated against every entity type and a final overall score is also given for all the measurements. Similarly, the Precision (PP), Recall (PR) and F1 measure (PF) for partial boundary matching as described in section 3.2 are also calculated. From the results depicted in Table 2 we can derive the following conclusions:

a) Loose boundary matching shows better results than the exact matching for every entity type across all the NER tools. That means there exist quite a number of cases where NER systems detected the right entity types but the boundaries are not exactly matched.
b) ORGANIZATION appears to be the entity type which is more difficult for detecting for all the NER tools. This is proved by its lower scores compared with the PERSON and LOCATION types.
c) Stanford NER and spaCy generally show better performance in this data set for both exact matching and partial matching.

4 Configuration of Hybrid NER System

4.1 Hybrid NER System

We need to have a NER system which is able to recognize PERSON, LOCATION, ORGANIZATION as well as DATE for our research projects. Among the evaluated NER tools, we selected the Stanford NER and spaCy for the configuration of the proposed hybrid NER system. Both tools showed good scores in our previous evaluation and are able to identify DATE entity without any extra setting (Stanford NER 7-class model includes the DATE type).

Our first target domain of application is Wikipedia pages about Singapore. To construct the hybrid NER system, we simply combined the outputs of the Stanford NER system and spaCy NER by using union method. In addition, a dictionary with limited entries on PERSON, LOCATION and ORGANIZATION about Singapore was also created with the expectation of improving system precision (Tsuruoka and Tsujii 2003; Cohen and Sarawagi, 2004). We set the dictionary to have the highest priority when there is any conflict with the outputs from other tools. Then followed by Stanford NER tool, it has the second highest priority on the determination of final named entities.

4.2 Data for Evaluation

In order to evaluate the performance of the hybrid system, we manually annotated twenty two web pages. All the web pages are from Singapore National Library Board eResources[3]. Half of the web pages are about Singapore history, another half are from Infopedia pages. We first use Stanford tool to tokenize all the documents and save them into different files. Every token is in a new line with a space line to separate the sentence. Then every token is manually annotated in IOB format. Table 3 shows the statistics of the two manually annotated datasets.

Entity Type	PER	LOC	ORG	DATE	Total
History	108	158	103	161	530
Infopedia	94	158	121	250	623

Table 3. Entity statistics on History and Infopedia testing datasets

When applying the same evaluation metrics as defined in section 3.2, we have the results on History data and Infopedia data as shown in table 4 and 5 respectively.

		PER	LOC	ORG	DATE	OVER ALL
Stanford	P	0.8649	0.8759	0.7527	0.7000	0.8004
	R	0.8889	0.7595	0.6796	0.5652	0.7113
	F	0.8767	0.8136	0.7143	0.6254	0.7532
	PP	0.8829	0.9270	0.8065	1.0000	0.9130
	PR	0.9074	0.8038	0.7282	0.8075	0.8113
	PF	0.8950	0.8610	0.7654	0.8935	0.8592
spaCy	P	0.7500	0.7889	0.3303	0.7407	0.6479
	R	0.6389	0.4494	0.3495	0.6211	0.5208
	F	0.6900	0.5726	0.3396	0.6756	0.5774
	PP	0.9022	0.9000	0.6055	0.9704	0.8474
	PR	0.7685	0.5127	0.6408	0.8137	0.6811
	PF	0.8300	0.6533	0.6227	0.8852	0.7552
Hybrid	P	0.8673	0.8212	0.7203	0.7962	0.8015
	R	0.9074	0.7848	0.8252	0.7764	0.8151
	F	0.8869	0.8026	0.7692	0.7862	0.8082
	PP	0.8761	0.8874	0.7458	0.9809	0.8813
	PR	0.9167	0.8481	0.8544	0.9565	0.8962
	PF	0.8959	0.8673	0.7964	0.9685	0.8887

Table 4. Evaluation results on History testing dataset

		PER	LOC	ORG	DATE	OVER ALL
Stanford	P	0.8500	0.8701	0.7080	0.7208	0.7819
	R	0.9043	0.8481	0.6612	0.5680	0.7079
	F	0.8763	0.8590	0.6838	0.6353	0.7431
	PP	0.8700	0.9091	0.7699	1.0000	0.9060
	PR	0.9255	0.8861	0.7190	0.7880	0.8202
	PF	0.8969	0.8975	0.7436	0.8814	0.8610
spaCy	P	0.6095	0.8917	0.2846	0.8551	0.6901
	R	0.6809	0.6772	0.2893	0.7080	0.6148
	F	0.6432	0.7698	0.2869	0.7746	0.6503
	PP	0.6952	0.9250	0.5610	1.0000	0.8288
	PR	0.7766	0.7025	0.5702	0.8280	0.7384
	PF	0.7336	0.7985	0.5656	0.9059	0.7810
Hybrid	P	0.7179	0.8187	0.5706	0.8826	0.7636
	R	0.8936	0.8861	0.7686	0.8120	0.8347
	F	0.7962	0.8511	0.6550	0.8458	0.7976
	PP	0.7436	0.8889	0.6196	1.0000	0.8370
	PR	0.9255	0.9620	0.8347	0.9200	0.9149
	PF	0.8246	0.9240	0.7112	0.9583	0.8742

Table 5. Evaluation results on Infopedia testing dataset

From the evaluation results on the History and Infopedia datasets, we can have the following remarks:

a) All the conclusions we drew from evaluation results over WikiGold dataset are still valid for the two manually annotated datasets: History and Infopedia.
b) Stanford NER generally shows good performance on all tested datasets. However, its scores on DATE entity type are not as good as spaCy. After further analysis on the false alarm and missing errors, we noticed that Stanford NER has difficulty to identify the full date information from the text. For instance, from text "*on 1 February 1858*", it can only identify "*February 1858*", the date is always missing. This problem is probably caused by the fact that Stanford NER is not trained for the date format "date month year". An alternative solution is to use its rule-based Temporal Tagger (SUTime). However, this is not included in the current evaluation.
c) The hybrid system usually has lower precision and higher recall than Stanford NER for entity types: PERSON, LOCATION, and ORGANIZATION. Its F1-measure is slightly better than Stanford

[3] http://eresources.nlb.gov.sg/index.aspx

NER for History data for these three entity types, but slight worse for Infopedia data.

d) In general, the hybrid system has better overall performance over both Stanford NER and spaCy. This is especially true for History testing data. However, most of the advantages are contributed by its better DATE entity recognition.

e) Overall, all the NER tools, including the hybrid system, showed better performance on History data than Infopedia data. This is mostly caused by some noise present in the Infopedia documents, for instance, html codes: *&rsquo;*, un-delimited words "*COMPASS.FamilyWife*" in the document due to the data extraction from the html pages.

5 Related Work

Different NER systems have been developed in the community and a number of them are freely available in the form of downloadable source codes/executables, web services or application programming interface for research purpose or limited use. Although these NER tools may differ in targeting domains, supported languages, processing methodologies, recognized entity types, and input/output formats, they can be evaluated in one way or another by applying the same evaluation metrics, such as traditional precision and recall. Marrero et al. (2009) evaluated ten NER tools which are targeting for general domains and English language. A small test corpus containing 579 English words are used for validation and observed that the variety of entity types that the tools can recognize does not determine the results. Atdag and Labatut (2013) compared four NER tools which include Stanford NER, Illinois NET, OpenCalais NER WS and Alias-i LingPipe for biographical texts. They created and annotated a new corpus from 247 Wikipedia articles and assessed their performance. They concluded that the testing results show a clear hierarchy between the tested tools: first Stanford NER, then LingPipe, Illinois NET and finally OpenCalais. Their results agree with our testing results for the two selected common NER tools. Kepa et al. 2012 evaluated the efficacy of four NER tools (OpenNLP, Stanford NER, AlchemyAP and OpenCalais) at extracting entities directly from the output of an optical character recognition (OCR) workflow. Their experiments showed that Stanford NER gave overall the best performance across two datasets, and was most effective on PER and LOC types. Alchemy API achieved the best results for the ORG type. In this paper, our work is different from the above mentioned validation tasks in the following ways:

a) We developed a validation framework which can work with various NER tools regardless of their programming languages. All the tools can work dynamically for immediate validation against gold standard corpus. The comparing results can be presented in text document or directly exported to excel file in predefined table format.

b) The selected tools are evaluated with both publicly available gold standard corpus and our manually annotated datasets.

c) After evaluating the selected NER tools, a further step was taken by combining the best performing NER tools in an effort to construct a new hybrid NER tool for our application domain.

6 Conclusion

In this paper, we conducted a comparative evaluation of four publically available and well-established NER tools which include Stanford NER, spaCy, Alias-i LingPipe and NLTK. For validation purposes, a framework has been developed in python, which can seamlessly work with different NER systems implemented in different programming languages. The output can be produced dynamically in both text documents or excel tables. The selected NER tools were evaluated by using publicly available gold standard corpus and our manually annotated datasets. Results showed that Stanford NER, followed by spaCy, performed the best across all the testing datasets. We further constructed a hybrid NER tool for our application domain by combining the best two performing NER tools.

In the future, we plan to continue improving the overall performance of the hybrid NER system by combining different features of more advanced systems as well as rule-based components.

Acknowledgements

We would like to thank Singapore National Library Board for providing all the original documents for our testing and validation.

Reference

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes,* 30:3-26.

Darvinder Kaur, Vishal Gupta. 2010. A survey of Name Entity Recognition in English and other Indian Langauges. *IJCSI International Journal of Computer Science,* Issues, Vol. 7, Issue 6.

Tjong Kim Sang, Erik. F., De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proc. Conference on Natural Language Learning*. pp. 142-147. Edmonton, Canada (2003).

Asma Ben Abacha, Pierre Zweigenbaum, 2001, Medical Entity Recognition: A Comparison of Semantic and Statistical Methods. *Proceedings of the 2011 Workshop on Biomedical Natural Language Processing*, ACL-HLT 2011, pages 56–64, Portland, Oregon, USA, June 23-24.

Lev Ratinov, Dan Roth, 2009. Design Challenges and Misconceptions in Named Entity Recognition, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 147–155, Boulder, Colorado, June 2009.

Rau, Lisa F. 1991. Extracting Company Names from Text. *In Proc. Conference on Artificial Intelligence Applications of IEEE.*

Collins, Michael; Singer, Y. 1999. Unsupervised Models for Named Entity Classification. *In Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Andrew Borthwick. 1999. Maximum Entropy Approach to Named Entity Recognition, *Ph.D. thesis, New York University*.

McCallum, Andrew; Li, W. 2003. Early Results for Named Entity Recognition with Conditional Random Fields, Features Induction and Web-Enhanced Lexicons. *In Proc. Conference on Computational Natural Language Learning*.

Takeuchi K. and Collier N. 2002. Use of Support Vector Machines in extended named entity recognition, *in the proceedings of the sixth Conference on Natural Language Learning (CoNLL-2002),* Taipei, Taiwan

Tim Rocktäschel, Michael Weidlich and Ulf Leser, 2012. ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics* 2012; 28:1633-40.

Srihari R., Niu C. and Li W. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging, *in the proceedings of the sixth Conference on Applied Natural Language Processing*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005),* pp. 363-370.

Alias-i. 2008. LingPipe 4.1.0. *http://alias-i.com/lingpipe* (accessed October 1, 2008).

Bird Steven, Ewan Klein, and Edward Loper, 2009. Natural Language, Processing with Python, *O'Reilly Media*.

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

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, James R. Curran, 2009. Named Entity Recognition inWikipedia, Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009, pages 10–18.

Grishman, Ralph and Sundheim, B. 1996. Message Understanding Conference - 6: A Brief History. *In Proc. International Conference on Computational Linguistics*.

Tsuruoka, Yoshimasa and Tsujii, J. 2003. Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. *In Proc. Conference of Association for Computational Linguistics. Natural Language Processing in Biomedicine*.

Cohen, William W., Sarawagi, S. 2004. Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods. *In Proc. Conference on Knowledge Discovery in Data*.

M. Marrero, S. Sanchez-Cuadrado, J. Lara, and G. Andreadakis. 2009. Evaluation of Named Entity Extraction Systems. *Advances in Computational Linguistics, Research in Computing Science,* 41:47–58.

S. Atdag and V. Labatut, 2013. A Comparison of Named Entity Recognition Tools Applied to Biographical Texts. *CoRR abs/1308.0661*, 2013.

Kepa Joseba Rodriquez, Mike Bryant, Tobias Blanke, Magdalena Luszczynska, 2012. Comparison of Named Entity Recognition tools for raw OCR text, *Proceedings of KONVENS 2012* (LThist 2012 workshop), Vienna, September 21, 2012.