Performance Comparison of Crowdworkers and NLP Tools on Named-Entity Recognition and Sentiment Analysis of Political Tweets

Mona Jalal*, Kate K. Mays**, Lei Guo**, and Margrit Betke*

*Department of Computer Science, Boston University
**Division of Emerging Media Studies, Boston University
{jalal,kkmays,guolei,betke}@bu.edu

Abstract

We report results of a comparison of the accuracy of crowdworkers and seven Natural Language Processing (NLP) toolkits in solving two important NLP tasks, named-entity recognition (NER) and entity-level sentiment (ELS) analysis. We here focus on a challenging dataset, 1,000 political tweets that were collected during the U.S. presidential primary election in February 2016. Each tweet refers to at least one of four presidential candidates, i.e., four named entities. The groundtruth, established by experts in political communication, has entity-level sentiment information for each candidate mentioned in the tweet. We tested several commercial and open source tools. Our experiments show that, for our dataset of political tweets, the most accurate NER system, Google Cloud NL, performed almost on par with crowdworkers, but the most accurate ELS analysis system, TensiStrength, did not match the accuracy of crowdworkers by a large margin of more than 30 percent points.

1 Introduction

As social media, specially Twitter, takes on an influential role in presidential elections in the U.S., natural language processing of political tweets (Mohammad et al., 2015) has the potential to help with nowcasting and forecasting of election results as well as identifying the main issues with a candidate – tasks of much interest to journalists, political scientists, and campaign organizers (Farzindar and Inkpen, 2015). As a methodology to obtain training data for a machine learning system that analyzes political tweets, Sameki et al. (2016) devised a crowdsourcing scheme with variable crowdworker numbers based on the difficulty of the annotation task. They provided a dataset of tweets where the sentiments towards political candidates were labeled both by experts in political communication and by crowdworkers who were likely not domain experts. Sameki et al. (2016) revealed that crowdworkers can match expert performance relatively accurately and in a budget-efficient manner. Given this result, the authors envisioned future work in which groundtruth labels would be crowdsourced for a large number of tweets and then used to design an automated NLP tool for political tweet analysis.

The question we address here is: How accurate are existing NLP tools for political tweet analysis? These tools would provide a baseline performance that any new machine learning system for political tweet analysis would compete against. We here explore whether existing NLP systems can answer the questions ***"What sentiment?"*** and ***"Towards whom?"*** accurately for the dataset of political tweets provided by Sameki et al. (2016). In our analysis, we include NLP tools with publicly-available APIs, even if the tools were not specifically designed for short texts like tweets, and, in particular, political tweets.

Our experiments reveal that the task of entity-level sentiment analysis is difficult for existing tools to answer accurately while the recognition of the entity, here, which politician, was easier.

2 NLP Toolkits

NLP toolkits typically have the following capabilities: tokenization, part-of-speech (PoS) tagging, chunking, named entity recognition and sentiment analysis. In a study by Pinto et al. (2016), it is shown that the well-known NLP toolkits NLTK (Bird and Loper, 2004), Stanford CoreNLP (Manning et al., 2014), and TwitterNLP (Ritter et al., 2011) have tokenization, PoS tagging and NER modules in their pipelines.

There are two main approaches for NER: (1) rule-based and (2) statistical or machine learning based. The most ubiquitous algorithms for sequence tagging use Hidden Markov Models (Jurafsky and Martin, 2009), Maximum Entropy Markov Models (Jurafsky and Martin, 2009; McCallum et al., 2000), or Conditional Random Fields (Sutton and McCallum, 2012). Recent works (Wang et al., 2016; Zhang and Liu, 2017) have used recurrent neural networks with attention modules for NER.

Sentiment detection tools like SentiStrength (Thelwall et al., 2010) and TensiStrength (Thelwall, 2017) are rule-based tools, relying on various dictionaries of emoticons, slangs, idioms, and ironic phrases, and set of rules that can detect the sentiment of a sentence overall or a targeted sentiment. Given a list of keywords, TensiStrength (similar to SentiStrength) reports the sentiment towards selected entities in a sentence, based on five levels of relaxation and five levels of stress.

Among commercial NLP toolkits (e.g., AYLIEN; MS Text Analytics; IBM Watson NLU), we selected Google Cloud Natural Language and Rosette Text Analytics for our experiments, which, to the best of our knowledge, are the only publicly accessible commercial APIs for the task of entity-level sentiment analysis that is agnostic to the text domain. We also report results of TensiStrength (Thelwall, 2017), TwitterNLP (Ritter et al., 2011), spaCy, CogComp-NLP (Khashabi et al., 2018), and Stanford NLP NER (Finkel et al., 2005).

3 Dataset and Analysis Methodology

We used the 1,000-tweet dataset by Sameki et al. (2016) that contains the named-entities labels and entity-level sentiments for each of the four 2016 presidential primary candidates Bernie Sanders, Donald Trump, Hillary Clinton, and Ted Cruz, provided by crowdworkers, and by experts in political communication, whose labels are considered groundtruth. The crowdworkers were located in the US and hired on the Amazon Mechanical Turk platform. For the task of entity-level sentiment analysis, a 3-scale rating of "negative," "neutral," and "positive" was used by the annotators.

Sameki et al. (2016) proposed a decision tree approach for computing the number of crowdworkers who should analyze a tweet based on the difficulty of the task. Tweets are labeled by 2, 3, 5, or 7 workers based on the difficulty of the task and the level of disagreement between the crowdworkers. The model computes the number of workers based on how long a tweet is, the presence of a link in a tweet, and the number of present sarcasm signals. Sarcasm is often used in political tweets and causes disagreement between the crowdworkers. The tweets that are deemed to be sarcastic by the decision tree model, are expected to be more difficult to annotate, and hence are allocated more crowdworkers to work on.

We conducted two sets of experiments. In the first set, we used TensiStrength, Google Cloud Natural Language, and Rosette Text Analytics, for entity-level sentiment analysis; in the second set, Google Cloud Natural Language, spaCy, Stanford NER, CogComp-NLP, and TwitterNLP, Rosette Text Analytics for named-entity recognition.

In the experiments that we conducted with TwitterNLP for named-entity recognition, we worked with the default values of the model. Furthermore, we selected the 3-class Stanford NER model, which uses the classes "person," "organization," and "location" because it resulted in higher accuracy compared to the 7-class model. For CogComp-NLP NER we used Ontonotes 5.0 NER model (Weischedel et al., 2013). For spaCy NER we used the 'en_core_web_lg' model.

We report the experimental results for our two tasks in terms of the correct classification rate (CCR). For sentiment analysis, we have a three-class problem (positive, negative, and neutral), where the classes are mutually exclusive. The CCR, averaged for a set of tweets, is defined to be the number of correctly-predicted sentiments over the number of groundtruth sentiments in these tweets. For NER, we consider that each tweet may reference up to four candidates, i.e., targeted entities. The CCR, averaged for a set of tweets, is the number of correctly predicted entities (candidates) over the number of groundtruth entities (candidates) in this set.

4 Results and Discussion

The dataset of 1,000 randomly selected tweets contains more than twice as many tweets about Trump than about the other candidates. In the named-entity recognition experiment, the average CCR of crowdworkers was 98.6%, while the CCR of the automated systems ranged from 77.2% to 96.7%. For four of the automated systems, detecting the entity Trump was more difficult than the other entities (e.g., spaCy 72.7% for the entity

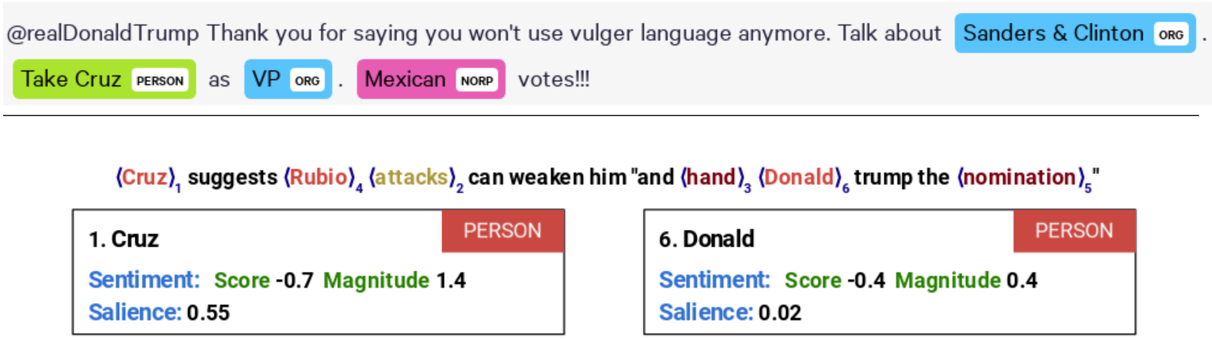

Figure 1: Incorrect NER by spaCy (top) and incorrect ELS analysis by Google Cloud (bottom)

Table 1: Average Correct Classification Rate (CCR) for named-entity recognition (NER) of four presidential candidates and entity-level sentiment (ELS) analysis by NLP tools and crowdworkers

		All Entities	Bernie Sanders	Donald Trump	Hillary Clinton	Ted Cruz
	# Tweets	1,000	236	510	225	211
NER	Rosette Text Analytics	**77.2%**	80.1%	79.8%	84.4%	60.2%
	TwitterNLP	**81.2%**	89.4%	76.5%	79.6%	85.3%
	CogComp-NLP	**82.6%**	83.9%	81.2%	83.9%	79.6%
	Stanford NER	**83.2%**	86.4%	76.8%	86.2%	83.4%
	spaCy	**88.2%**	91.1%	72.7%	92%	91.5%
	Google Cloud NL	**96.7%**	97.9%	96.1%	96.4%	97.1%
	mTurk crowdworkers	**98.6%**	100%	99%	97.3%	98.1%
ELS	Rosette Text Analytics	**31.7%**	38.5%	24.9%	40.4%	31.3%
	Google Cloud NL	**43.2%**	44.9%	40.8%	45.5%	44.8%
	TensiStrength	**44.2%**	52.1%	40.8%	43.6%	44.1%
	mTurk crowdworkers	**74.7%**	77.9%	71.7%	67.5%	80.5%

Trump vs. above 91% for the other entities). An example of incorrect NER is shown in Figure 1 top. The difficulties the automated tools had in NER may be explained by the fact that the tools were not trained on tweets, except for TwitterNLP, which was not in active development when the data was created (Farzindar and Inkpen, 2015).

In the sentiment analysis experiments, we found that a tweet may contain multiple sentiments. The groundtruth labels contain 210 positive sentiments, 521 neutral sentiments, and 305 negative sentiments to the candidates. We measured the CCR, across all tweets, to be 31.7% for Rosette Text Analytics, 43.2% for Google Cloud, 44.2% for TensiStrength, and 74.7% for the crowdworkers. This means the difference between the performance of the tools and the crowdworkers is significant – more than 30 percent points.

Crowdworkers correctly identified 62% of the neutral, 85% of the positive, and 92% of the negative sentiments. Google Cloud correctly identified 88% of the neutral sentiments, but only 3% of the positive, and 19% of the negative sentiments. TensiStrength correctly identified 87.2% of the neutral sentiments, but 10.5% of the positive, and 8.1% of the negative sentiments. Rosette Text Analytics correctly identified 22.7% of neutral sentiments, 38.1% of negative sentiments and 40.9% of positive sentiments. The lowest and highest CCR pertains to tweets about Trump and Sanders for both Google Cloud and TensiStrength, Trump and Clinton for Rosette Text Analytics, and Clinton and Cruz for crowdworkers. An example of incorrect ELS analysis is shown in Figure 1 bottom.

5 Conclusions and Future Work

Our results show that existing NLP systems cannot accurately perform sentiment analysis of political tweets in the dataset we experimented with. Labeling by humans, even non-expert crowdworkers, yields accuracy results that are well above the results of existing automated NLP systems. In future work we will therefore use a crowdworker-labeled dataset to train a new machine-learning based NLP system for tweet analysis. We will ensure that the training data is balanced among classes. Our plan is to use state-of-the-art deep neural networks and compare their performance for entity-level sentiment analysis of political tweets.

Acknowledgments

Partial support of this work by the Hariri Institute for Computing and Computational Science & Engineering at Boston University (to L.G.) and a Google Faculty Research Award (to M.B. and L.G.) is gratefully acknowledged. Additionally, we would like to thank Daniel Khashabi for his help in running the CogComp-NLP Python API and Mike Thelwal for his help with TensiStrength. We are also grateful to the Stanford NLP group for clarifying some of the questions we had with regards to the Stanford NER tool.

References

Amazon Mechanical Turk. https://www.mturk.com. Last accessed on May 12, 2018.

AYLIEN. https://developer. aylien.com/text-api-demo. Last accessed on March 07, 2018.

Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26, 2004 - Poster and Demonstration*.

CogComp-NLP. http://nlp.cogcomp.org. Last accessed on May 17, 2018.

Atefeh Farzindar and Diana Inkpen. 2015. Natural language processing for social media. *Synthesis Lectures on Human Language Technologies*, 8(2):1–166.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 363–370.

Google Cloud Natural Language. https://cloud. google.com/natural-language. Last accessed on March 09, 2018.

IBM Watson NLU. IBM Watson Natural Language Understanding. https://www. ibm.com/watson/services/natural-language-understanding. Last accessed on March 07, 2018.

Dan Jurafsky and James H. Martin. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, Second Edition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International.

Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev Ratinov, Guanheng Luo, Quang Do, Chen-Tse Tsai, Subhro Roy, Stephen Mayhew, Zhilli Feng, John Wieting, Xiaodong Yu, Yangqiu Song, Shashank Gupta, Shyam Upadhyay, Naveen Arivazhagan, Qiang Ning, Shaoshi Ling, and Dan Roth. 2018. Cogcompnlp: Your swiss army knife for nlp. In *International Conference on Language Resources and Evaluation (LREC)*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 591–598.

Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel D. Martin. 2015. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499.

MS Text Analytics. https://azure.microsoft.com/en-us/services/ cognitive-services/text-analytics, Last accessed on March 07, 2018.

Alexandre Miguel Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. 2016. Comparing the performance of different NLP toolkits in formal and social media text. In *5th Symposium on Languages, Applications and Technologies, SLATE 2016, June 20-21, 2016, Maribor, Slovenia*, pages 3:1–3:16.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in Tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1524–1534.

Rosette Text Analytics. https://www.rosette.com. Last accessed on May 18, 2018.

Mehrnoosh Sameki, Mattia Gentil, Kate K. Mays, Lei Guo, and Margrit Betke. 2016. Dynamic allocation of crowd contributions for sentiment analysis during the 2016 U.S. presidential election. In *The Fourth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2016), October 30-November 3, 2016. 10 pages.*

spaCy. http://spacy.io Last accessed on May 16, 2018.

Stanford NER. https://nlp.stanford.edu/software/crf-ner.shtml. Last accessed on May 17, 2018.

Charles A. Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

TensiStrength. http://sentistrength.wlv.ac.uk/ tensistrength.html. Last accessed on March 09, 2018.

Mike Thelwall. 2017. TensiStrength: Stress and relaxation magnitude detection for social media texts. *Information Processing & Management*, 53(1):106–121.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology JASIST*, 61(12):2544–2558.

TwitterNLP. https://github.com/aritter/twitter_nlp. Last accessed on May 17, 2018.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.

Yue Zhang and Jiangming Liu. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, April 3-7, 2017*, volume 2, pages 572–577.