Stance Classification Towards Political Figures on Blog Writing

Conference Paper · November 2018

DOI: 10.1109/IALP.2018.8629144

Stance Classification towards Political Figures on Blog Writing

Rini Jannati[1], Rahmad Mahendra[2], Cakra Wishnu Wardhana[3] and Mirna Adriani[4]
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
[1]*rini.jannati@ui.ac.id,*[3] *cakra.wishnu@gmail.com,* {[2]*rahmad.mahendra,* [4]*mirna*}*@cs.ui.ac.id*

***Abstract*—In this paper, we present new application of stance detection task on politic domain. Our goal is to determine whether the writer of the blog article is on the position supporting a political figure to compete and win in a general election event, for example a candidate of President in the Presidential election. We performed the experiment using five different case studies.We examined three baseline machine learning models using combination of n-gram, sentiment lexicon, orthography, and word embedding features. The highest macro-average F1 score was achieved by model trained on Support Vector Machine classifier using a combination of word2vec and unigram features, which is 63,54%.**

Keywords-Stance Detection; Political Text Analytics; Text Classification; Blog

I. Introduction

Social webs nowadays have been inseparable aspects of political activities. During election time, those are vital campaign tools. The politicians and political parties raise some issues in the social media. They express their political agenda online and broadcast them to millions of people instantaneously in almost real time. The political figures brand their images via internet in order to influence the public to support and vote them.

On the other hand, more and more people talk and discuss political contents in the blogsphere. The conversation about the politicians and political parties are prevalent. The latest news about the candidates during election often become the trending topics. Analyzing abundant online posts in politics allows the politicians to carefully manage their campaign. By grasping the insights from the posts, they learn why their supporters keep loving them and what their haters dislike from them, and more importantly, which concerns are being considered by the swing voters to vote them or their rivals.

With the explosion of Web and increased activity in blogging, tagging, and commenting, there has been growing interest to mine these vast resources for opinions. Sentiment analysis is the computational process to identify and categorize opinions expressed in a piece of text, in order to determine whether the writer's attitude towards a particular topic is positive, negative, or neutral. Sentiment analysis task has been performed in a number of works in political domain, including but not limited to, Tumasjan [1], Wang [2], Bakliwal [3], DiFatta [4], Ibrahim [5], and Kuen [6]. Analyzing sentiment in political text can be proxy of election outcome, measurement of candidate electability, or representation of real time political landscape in a nation.

Election is recurrent seasonal event, in which a number of different names come and go. A candidate wins a Presidential or Gubernatorial election in particular year, serves for several years in his term of office,and then competes as incumbents in the following election event. Another candidate loses this one and wins in the future. From the public perspective itself, the sentiment towards political situation is not rigid. For example, how a politician is evaluated depends much on situational context. Whether he or she is currently a candidate of ongoing election, who is the opponent, which parties are supporting him or her, are few example of assesments.

In this work, we examine how to categorize textual passages in the social web that instantiate a relevant position of its writer in a specific political event. To be more specific, we use a collection of blog articles to detect the stance towards a candidate in a particular Presidential of Gubernatorial election.

The rest of paper is organized as follows. We first explain the stance detection task in Section II. Our task definition and method are described in Section III. The result of experiments and analysis are presented in Section IV. Finally, conclusions are drawn and future work is mentioned in last section.

II. Related Work

Stance is one's position or standing points towards a target, i.e. person, objects, ideas, or opinions [7]. The position can be Favor (e.g., directly or indirectly by supporting someone/something, by opposing or criticizing someone/something opposed to the target, or by echoing the stance of somebody else), Against (e.g., directly or indirectly by opposing or criticizing someone/something, by supporting someone/something opposed to the target, or by echoing the stance of somebody else); or neither (e.g. standing in neutral position, or not clearly expressing the stance within the passage). Stance detection is the task of automatically determining from the text whether the author of the text is in favor of, against, or neutral towards a proposition or target. [7] [8]. Stance detection is much similar to sentiment analysis task. Opinion sentiment can be used to identify the stance of the document. However, the sentiment category does not simply reflect the stance.

The document containing positive sentiment does not always stand for favor stance [9].

Previous work on stance detection mostly used the data from debates, e.g. Congress debates [10] or debates in online forum [11], and conversation about controversial issues in social media, e.g. Twitter [12].

Anand et al [13] experimented with 4K posts from 14 topics. The goal of their work was to classify whether a post is rebuttal or not. They proposed several features to identify the stance, e.g. cue words, repeated punctuations, and syntactic dependencies. Hasan and Ng [10] examined several factors in a stance classification task. They found that adding training data contributes to improve system performance. Besides that, n-gram feature is still considered as one of most important feature for stance prediction in addition to other features, such as document statistic. They also compared Naive Bayes and SVM classifier, and concluded that no clear winner between two of them in classifying stance in text.

Sobhani et al [9] detected stance on the tweet dataset. In their study, the expression of the target was at a specified text using positive or negative language in tweets. The stance label was determined using model trained on linear-kernel Support Vector Machine (SVM) classifier using three features, i.e. n-gram, sentiment, and word embedding. This simple model achieved F-Score 70.32, which outperformed more complex model provided by the best team in SemEval 2016 task competition.

Faulker [14] applied stance classification task to automatically detect the argument stance of a student essay given a discussion topic. Sasaki et al [15] worked on stance detection towards an event. To classify the stance, they recognized a situation in which the event occurs or does not occur.

Politics is one of frequent topic chosen as a case study of stance detection task. For example, Hillary Clinton and Donald Trump topic in Mohammad [16]. Most recent work by Sobhani et.al. [17] annotated multi-target stance data, which is a number of pair of Presidential candidate for US election 2016.

III. METHOD

In our work, the problem of stance detection towards the political figures is approached as a binary text classification task. Suppose that we have an input of a person entity P_i, an event E_j, and a document D_k, the goal of task is to find Y_l that maximize the probability value $P(Y_l|(P_i, E_j, D_k))$, where $y_i \in$ {favor, against}. If the output of classifier is Y_l ="favor", it means that the document D_k is in favor of the person entity P_i in context of the event E_j. In our task, P_i and E_j are respectively restricted to a candidate of President / Governor, and Presidential / Gubernatorial election. The documents D in our experiment are the collection of blog posts. So, Y_l ="favor" can be interpreted that a blog post D_k expresses the writer‘s position in supporting a candidate P_i in the context of an election event E_j.

To conduct this work, we followed systematic methodology, including data collection, data annotation, feature engineering, model building, and evaluation.

A. Dataset Preparation

The data used in this study was the corpus of articles from Kompasiana[1]. Any online user can write various topics in Kompasiana, including politics. The article typically consists of few paragraphs. Even though the writing style mostly resembles blog post, the language being used in most articles is formal and standard. The language is grammatical and uses proper punctuations.

We collected the articles using pre-defined keywords. The keywords were list of politician names that become input of model classifier. We did not only use the full name, but also extended the list with popular alias names of respective political figures. To ensure that the article corresponding to particular politician discusses the context of his / her candidacy for President or Governor, we considered the time when the article was written. We maintained the uniqueness of articles by removing the duplicate, including two or more articles that have different titles, but the contents are the same each other. We removed short articles whose length is less than 1000 characters.

The annotation was done in two stages. First, the annotators isolated the articles that does not expressing stance information in spite of discussing a candidate with context of Presidential or Gubernatorial election event. For each article possessing stance towards candidate entity and election event, the annotators determined the correct label. Since our task is binary classification, the articles with no stance were excluded from gold-standard set for the experimental purpose.

The experiments were conducted using five different case studies (i.e. five different pairs of candidate entity and election event), which were:

- Ahok for Gubernatorial Election DKI 2017
- Jokowi for Presidential Election 2014
- Prabowo for Presidential Election 2014
- Agus Harimukti Yudhoyono (AHY) for Presidential Election 2019
- Ridwan Kamil for Gubernatorial Election Jabar 2018

The number of annotated articles with stance label was 337. The detail is described in Tabel I.

TABLE I. STATISTIC OF GOLD-STANDARD DATASET

Entity	Event	Favor	Against	Total
Ahok	Gubernatorial Election DKI 2017	101	54	155
Jokowi	Presidential Election 2014	40	18	59
Prabowo	Presidential Election 2014	22	12	34
AHY	Presidential Election 2019	33	10	43
Ridwan Kamil	Gubernatorial Election Jabar 2018	17	29	46

[1]One of most popular and largest citizen journalism site in Indonesia,(www.kompasiana.com).

The stance of article towards target (pair of entity and event) can be found straightforwardly in explicit statements that were written by the author, usually in first or last paragraph. On the other hand, sometimes the stance should be inferred from implicit arguments provided within the text. Some articles mentioned a number of political figures, made the comparison among them, praised one figure and criticized the others.

The example of annotated article with favor label is as follows. The article supported Ahok (entity) on Gubernatorial Election for DKI Jakarta 2017 (event) because the arguments within article were constructed to against other candidates beside Ahok.

Entity: Ahok
Event: Gubernatorial DKI Jakarta 2017
Text:
Yusril tidak menyadari pernyataan tsb akan memberi kesan ia mengejar jabatan Gubernur DKI hanya sebagai batu loncatan untuk ikut Pilpres 2019. Apakah mungkin warga DKI mau memilih cagub dengan niat seperti itu? Apalagi niat tsb dinyatakannya dengan terang-terangan. Yusril perlu belajar dari Jokowi pada saat masih menjabat sebagai gubernur DKI ketika para wartawan beramai-ramai menanyakan niatnya untuk mengikuti Pilpres 2014, Jokowi menjawab: Gak mikir. Padahal tentunya diam-diam Jokowi pengen juga menjadi presiden. Siapa sih yang gak mau jadi presiden jika kesempatan tsb memang ada? Sudah pernah jadi menteri, lalu sekarang turun kelas ingin jadi gubernur, niat inipun semakin menampakkan Yusril hanya mau mengejar jabatan. Berbeda dengan Ahok yang ketika ditanya soal kesiapannya menghadapi Pligub 2014, Ahok menyatakan belum mau memusingkan strategi kampanye menyambut Pilgub DKI 2017. Menurut Ahok, saat ini dia ingin bekerja dengan baik agar saat selesai masa jabatan nanti ada yang bisa dikenang oleh warga Jakarta.Tanpa bermaksud memuji Ahok secara berlebihan, yang jelas pernyataan Ahok tsb telah mencerminkan dia lebih mementingkan pengabdian daripada sekedar mengejar jabatan.Di lain kesempatan Ahok pun menyatakan bahwa dirinya siap kalah pada Pilgub DKI 2017. Dengan kelemahan-kelemahan seperti itu tampaknya Yusril akan sangat sulit mengalahkan Ahok. Seperti telah diketahui hasil survey menunjukkan Ahok menempati posisi teratas dibandingkan bakal cagub DKI lainnya. Juru bicara Partai Demokrat Ruhut Sitompul menyatakan, Ahok sulit ditandingi (viva.co.id, 26-2-2015), karena itu jika Yusril merasa yakin bisa mengalahkan Ahok seandainya Pilgub DKI 2017 berlangsung head to head seperti Prabowo VS Jokowi di Pilpres 2014, maka mungkin saja ia juga akan keok seperti Prabowo. Semoga bermanfaat.
(Yusril was not aware with his statement would give the impression that he was pursuing the position of DKI Gubernatorial only as a stepping stone to participate in the 2019 Presidential Election. Is it possible that DKI residents would choose next governor with such intentions? Moreover, the intention was expressed openly. Yusril needs to learn from Jokowi while still serving as the governor of DKI, when journalists roll around asking for his intention to take part in the 2014 presidential election, Jokowi replied: I don't think. In fact, of course Jokowi secretly wants to become president. Who doesn't want to be president if the opportunity really exists? Having been a minister, and now going down the class to become a governor, even this intention shows Yusril only wanting to pursue office. Unlike Ahok, who when asked about his readiness to face the 2014 Gubernatorial Election, Ahok stated that he did not want to confuse the campaign strategy to 2017 Gubernatorial Election of DKI. Praising Ahok excessively, what is clear is that Ahok's statement has reflected that he is more concerned with dedication than just pursuing a position. On another occasion Ahok also stated that he was ready to lose to the Gubernatorial Election DKI. As already known the survey results show Ahok occupies the top position compared to other Gubernatorial Election of DKI. Democratic Party spokesman, Ruhut Sitompul, said Ahok is difficult to match (viva.co.id, 26-2-2015), therefore if Yusril feels confident of beating Ahok in case the Gubernatorial Election DKI 2016 takes place head to head like Prabowo VS Jokowi in the 2014 presidential election, then maybe he will also be as clumsy as Prabowo. May be useful.)

The following is an example of annotated article with against label. Unlike previous example, the author provided more explicit arguments to stand against Ridwan Kamil (entity) in the context of Gubernatorial Election for Jabar 2018 (event).

Entity: Ridwan Kamil
Event: Gubernatorial Election Jabar 2018
Text:
Hasilnya, di bawah kepemimpinan Ridwan Kamil, masalah utama Kota Bandung disebut gagal teratasi. Berdasarkan survei tersebut, ada tiga permasalahan utama yang justru tidak tertangani menurut masyarakat. Yakni ekonomi, kemacetan dan banjir. Ridwan kamil ini dinilai gagal (kinerja). Kenapa? Karena tidak sinkron antara apa yang diharapkan masyarakat dan apa yang dikerjakan pemkot Bandung. Jadi publik tidak mendukungnya maju sebagai cagub.
(The result of Ridwan Kamil era, the important problem in Bandung cant be resolved. Based on the survey, citizen said there are three im-

portant problem that cant be resolved. Ridwan Kamil was failed. Why? Because, there was not syncronization between citizen wished and what government of Bandung did. So, citizen will not support him as candidate of Governor.)

B. Feature Engineering

We applied four features for classification in this study.

1) N-Gram: The bag of word features in our work include unigram and bigram.

2) Sentiment Lexicon: This lexicon consists of the list of 415 positive words and 581 negative words [18]. The sentiment feature are the number of positive and negative terms within the document.

3) Orthography: This feature group includes the frequency of question marks, the frequency of exclamation marks, and the number of capitalized words that were not at the beginning of the sentence in the article.

4) Word Embedding: Word2Vec [19] vector representation of text is obtained from model that is trained by the collection of articles in politics domain.

C. Model Building

We experimented with three different classifiers to identify stance label for blog post.

1) Multinomial Naive Bayes: Multinomial Naive Bayes (MNB) is one of classic Naive Bayes variations for multinomial distribution data. It is commonly used as baseline for text classification [20].

2) Support Vector Machine: The basic idea of Support Vector Machine (SVM) classifier is find a hyperplane that separate the data into two class with the highest margin value. The margin election does not involve the whole element, but only the elements in positions that intersect with the margin. These elements are called support vector [21].

3) Logistic Regression: Logistic regression makes a connection between a categorical dependent variable and a set of independent (explanatory) variables. The model does not assume that the independent variables are normally distributed [22].

D. Evaluation

We applied 5-fold cross validation in our experimental setting. In each experiment, we evaluated the model in order to see how accurate it predicts the stance label as expected. The metrics for our evaluation were precision, recall, and F1-measure. These metrics are applied to all cases, which are pairs of political figures and events. The macro average precision, recall, and F1 of a model were computed to assess overall performance of the stance detector.

IV. Experiment And Result

The main objective of our experiment to check the which features are important and which models perform well in classifying stance toward political figures. The summary of experimental result is presented in Table II.

TABLE II. RESULT OF CLASSIFICATION

Features	**MNB**	**SVM**	**LR**
Single Feature			
N-Gram			
- Unigram(UNI)	56.17%	62.67%	60.06%
- Bigram(BI)	43.34%	55.02%	54.18%
- UNI + BI	56.05%	61.42%	59.90%
Sentiment Lexicon (SENT)	50.42%	48.24%	49.60%
Orthography (ORT)	46.10%	42.78%	44.15%
Word2Vec (VEC)	42.86%	57.57%	51.42%
Combination 2 Features			
UNI + SENT	58.75%	59.61%	59.99%
UNI + ORT	52.39%	41.86%	60.69%
UNI+ VEC	56.28%	**63.54%**	62.18%
SENT + ORT	43.15%	28.67%	51.90%
SENT + VEC	54.44%	55.46%	57.11%
ORT + VEC	47.49%	27.81%	47.93%
Combination more than 2 Features			
UNI + SENT + ORT	53.63%	43.42%	61.78%
UNI + SENT + VEC	*58.87%*	61.98%	60.33%
UNI + ORT + VEC	54.84%	43.82%	61.31%
SENT + ORT + VEC	52.50%	57.38%	57.04%
UNI + SENT + ORT + VEC	56.23%	40.73%	*62.34%*

Experiments with single features yielded the best result when using unigram, which gave the highest F1-measure 56.17%, 62.67%, and 60.06% using classifier model Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Logistic Regression (LR), respectively. When unigram was combined with bigram feature, the result was still lower than using unigram only. Overall, orthography feature is least important among four features.

Experiments using combination of two features contributed the highest F1-measure when the unigram was incorporated with word2Vec features. The second best was obtained by combination of unigram and sentiment lexicon features. The lowest F-measure were obtained by using orthography and word2Vec features.

When a combination of unigram and sentiment lexicon is classified with MNB, the result increased compared to the result of single features. On the other hand, with the SVM and LR classifiers, unigram feature still got higher score than combination of unigram and sentiment lexicon. Overall, F1-measure of unigram feature still outperformed F1-measure of any combination of two features. The only exception are combination of unigram and Word2Vec features trained on SVM model, achieved F1-measure 63.54%, higher than F1-measure of unigram using SVM which was 62.67%.

Different results are shown for sentiment lexicon and orthography features, in which when their combination with unigram feature was used, the F1-measure were increased compared to only using single feature. F1-measures of combination of unigram and sentiment lexicon were increased by 8.33%, 11.37%, 10.39% for MNB, SVM, and LR respectively, compared to using sentiment lexicon feature only. F-measures of unigram and orthography were also increased by 6.29% and 11.09% when using MNB and LR compared to only using orthography feature, but was decreased by 0.38% when using SVM. The combination of unigram and Word2Vec also increased F-measures by 11.58%, 5.97%, and 10.76% for MNB, SVM, and LR, compared to using only Word2Vec. The best results of combination of two features were obtained from unigram combined with other features.

The experiment using combination of more than two features results did not give significant improvements. The effect of feature combinations depends on the classifier. There are no combination of features that successfully increase score in all different classifiers. Some combinations offered the higher score using LR, but not with other classifiers.

Feature combinations used in this experiment have different effect on each classifier. When using MNB, the highest score is obtained when using a combination of unigram, sentiment lexicon and Word2Vec (58.87%). A combination of all features (unigram, sentiment lexicon, orthography and Word2Vec) has higher score than the combination of unigram and Word2Vec using LR (63.34%). This is different from SVM which has the highest score using the combination of unigram and Word2Vec (63.54%). The best F1-measures of all experiments is 63.54%, which was the result of using combination of unigram and Word2Vec features, with SVM as the classifier.

The detailed results of this scenario for five cases we used are displayed in Table III.

TABLE III. A Combination of Unigram and Word2Vec using SVM's Result

Entities	Events	Precision	Recall	F1
Ahok	Gubernatorial Election DKI 2017	69.66%	66.51%	67.90%
Jokowi	Presidential Election 2014	56.81%	56.28%	56.50%
Prabowo	Presidential Election 2014	79.00%	75.33%	77.03%
AHY	Presidential Election 2019	50.83%	57.50%	52.42%
Ridwan Kamil	Gubernatorial Election Jabar 2018	64.76%	63.14%	63.77%
Averages		**64.21%**	**63.75%**	**63.54%**

A case of Prabowo (entity) and Presidential Election 2014 (event) has higher F1 score than the others. Precision score is higher than recall on several cases (except AHYs case). Imbalance data is one of the reasons that F1 score is only around 50%-70%. Data distribution between favor and against label reaches a ratio of around 2:1 or 1:2. There will be a possibility that against being suspected as favor label, vice versa. It is because the classifier recognizes dominant label in the training data.Several cases consist of entities like Ahok, Jokowi and Prabowo contain other entity that the case is compared with entity (like implicitly support entity but explicitly againts other entity). But in AHYs case is different. Many articles with against label has similarity with favor label. The article labeled as Against in AHYs case contains words that support that figure for other events.

V. Conclusion And Future Work

This study aims to detect stance in the blog article towards political figures. The target consists of entity (political figures, i.e. Presidential or Gubernatorial candidate) and event (election). We tested several features combinations using three different machine learning models, such as Multinomial Naive Bayes (MNB), Linear Support Vector Machine (SVM), and Logistic Regression. The highest macro-average F1-score in this study was 63,54% using SVM with a combination of unigram and Word2Vec.

Unigram feature was considerably most important feature. Combining unigram with word embedding feature, i.e. Word2Vec, can improve F1-measure on each classifier. Imbalanced data affects the classification results. Some articles contain multiple entities and/or events, which were still difficult to determine the stance toward particular entity and event. Some other articles do not provide explicit statement to express the stance, so the lexical features are still not enough to solve this problem.

For the future work, we plan to implement this task as two-stage classification for full pipeline (another model to automatically detect the article with stance may be designed). The class label can also be expanded, for example discussed, neutral, and mixed stances.

Acknowledgment

This work is supported by Hibah PITTA 2018 funded by DRPM Universitas Indonesia No. 1885/UN2.R3.1/HKP.05.00/2018.

References

[1] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, 2010, pp. 178–185.

[2] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*, ser. ACL '12, 2012, pp. 115–120.

[3] A. Bakliwal, J. Foster, J. van der Puil, R. O'Brien, L. Tounsi, and M. Hughes, "Sentiment analysis of political tweets: Towards an accurate classifier," in *Proceedings of the Workshop on Language Analysis in Social Media*, 2013, pp. 49–58.

[4] G. Di Fatta, J. J. Reade, S. Jaworska, and A. Nanda, "Big social data and political sentiment: The tweet stream during the UK general election 2015 campaign," *Proceedings - 2015 IEEE International Conference on Smart City, SmartCity 2015*, pp. 293–298, 2015.

[5] M. Ibrahim, O. Abdillah, A. F. Wicaksono, and M. Adriani, "Buzzer Detection and Sentiment Analysis for Predicting Presidential Election Results in a Twitter Nation," *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, pp. 1348–1353, 2016.

[6] E. Kuen and M. Strembeck, "Politics, sentiments, and misinformation: An analysis of the twitter discussion on the 2016 austrian presidential elections," *Online Social Networks and Media*, vol. 5, pp. 37 – 50, 2018.

[7] S. Somasundaran and J. Wiebe, "Recognizing stances in ideological on-line debates," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, ser. CAAGET '10, 2010, pp. 116–124.

[8] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, "Stance and Sentiment in Tweets," *ACM Transactions on Internet Technology (TOIT) - Special Issue on Argumentation in Social Media and Regular Papers*, vol. 17, no. 3, 2016.

[9] P. Sobhani, S. M. Mohammad, and S. Kiritchenko, "Detecting Stance in Tweets And Analyzing its Interaction with Sentiment," *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*Sem)*, pp. 159–169, 2016.

[10] K. S. Hasan and V. Ng, "Stance Classification of Ideological Debates : Data , Models , Features , and Constraints," *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 1348–1356, 2013.

[11] M. A. Walker, P. Anand, R. Abbott, J. E. F. Tree, C. Martell, and J. King, "That is your evidence?: Classifying stance in online political debate," *Decision Support Systems*, vol. 53, no. 4, pp. 719 – 729, 2012.

[12] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, 2016, pp. 31–41. [Online]. Available: http://aclweb.org/anthology/S16-1003

[13] P. Anand, M. Walker, R. Abbott, J. E. F. Tree, R. Bowmani, and M. Minor, "Cats Rule and Dogs Drool!: Classifying Stance in Online Debate," *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pp. 1–9, 2011.

[14] A. Faulkner, "Automated Classification of Stance in Student Essays: An Approach Using Stance Target Information and the Wikipedia Link-Based Measure," *Proceedings of the 27th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014*, vol. 376, no. 12, pp. 174–179, 2014.

[15] A. Sasaki, J. Mizuno, N. Okazaki, and K. Inui, "Stance Classification by Recognizing Related Events about Targets," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2016, pp. 582–587.

[16] S. Mohammad, S. Kiritchenko, P. Sobhani, X.-D. Zhu, and C. Cherry, "A dataset for detecting stance in tweets," in *LREC*, 2016.

[17] P. Sobhani, D. Inkpen, and X. Zhu, "A dataset for multi-target stance detection," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 551–557.

[18] C. Vania, M. Ibrahim, and M. Adriani, "Sentiment Lexicon Generation for an Under-Resourced Language," *International Journal of Computational Linguistics and Applications*, vol. 5, no. 1, pp. 59–72, 2014.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*. Curran Associates, Inc., 2013, pp. 3111–3119.

[20] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Learning for Text Categorization: Papers from the 1998 AAAI Workshop*, 1998, pp. 41–48.

[21] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[22] D. W. Hosmer and S. Lemeshow, *Applied logistic regression*, 2nd ed. Wiley-Interscience Publication, Sep. 2000.