Towards Detection of Influential Sentences Affecting Reputation in Wikipedia

Yiwei Zhou
Department of Computer Science
University of Warwick
Coventry, UK
Yiwei.Zhou@warwick.ac.uk

Alexandra I. Cristea
Department of Computer Science
University of Warwick
Coventry, UK
A.I.Cristea@warwick.ac.uk

ABSTRACT

Wikipedia has become the most frequently viewed online encyclopaedia website. Some sentences in Wikipedia articles have direct and obvious impact on people's opinions towards the mentioned named entities. This paper defines and tackles the problem of *reputation-influential* sentence detection in Wikipedia articles from various domains. We leverage multiple lexicons, to generate *domain independent features*. We generate *topical features* and *word embedding features* from unlabelled dataset, to boost the classification performance. We conduct several experiments, to prove the effectiveness of these features. We further adapt a *two-step binary classification method*, to perform multi-classification. Our evaluation results show that this method outperforms the state-of-the-art one-vs-one multi-classification method for this problem.

CCS Concepts

•**Information systems** → **Web mining;** •**Computing methodologies** → *Natural language processing;*

Keywords

Wikipedia; Cross-domain classification; Reputation-influential

1. INTRODUCTION

Wikipedia has become one of the most frequently used websites in people's daily life. Take just the English Wikipedia for an example, it contains more than 5 million articles and receives more than 5 million views per hour[1]. Such comprehensive information inclusion and huge visiting traffic make Wikipedia influential for people all around the world. Due to the NPOV[2] policy, most sentences in Wikipedia are unopinionate and unbiased. However, Wikipedians manage to implicitly express their opinions, by including selective facts and varying description patterns, which we have shown to lead to bias for **events** at the *article level* [21] and bias for **named entities** at the *corpus level* [22]. Some sentences on Wikipedia, even though they just state some facts, or they come from reliable sources, have strong influence on Wikipedia users' opinions about the named entities mentioned in them. For example, sentences in Wikipedia like "Chevron did not apologise, nor paid the amount of compensation." and "There are some exceptions, such as striker Wayne Rooney, who became extremely unpopular with fans after changing Everton for Manchester United, and is currently always booed when he returns to the stage of his former club." would negatively impact their mentioned entities' reputation, which are "Chevron Corporation" and "Wayne Rooney". Sentences in Wikipedia like "Lady Gaga won two awards, including the prize for best song for Born This Way at the Europe Music Awards." and "Boeing today is a synonymous name for dynamic, impressive aircraft, global air travel, success and economic strength." positively impact their mentioned entities' reputation, which are "Lady Gaga" and "Boeing Company". We call this kind of sentences *reputation-influential* sentences. If a sentence can stimulate positive opinions towards the mentioned named entity, then it is a *positive reputation-influential* sentence; if a sentence can stimulate negative opinions towards the mentioned named entity, then it is a *negative reputation-influential* sentence.

This paper aims at the detection of positive and negative *reputation-influential* sentences from Wikipedia articles. This is **not** a traditional sentiment analysis problem, as the sentiments are only implicitly expressed or even hidden in Wikipedia sentences. However, they have positive or negative implications for the mentioned named entities' reputation, and can influence people's opinions towards them implicitly. To the best of our knowledge, this is the first paper to define such a problem for Wikipedia sentences.

We apply a two-step binary classification method, as explained in Section 3, to tackle this cross-domain multi-classification problem on Wikipedia sentences. We use multiple lexicons, as mentioned in Section 3.1, to generate domain independent features. Because of the lack of large annotated datasets from various domains, we generate unsupervised features from our unlabelled dataset. Our evaluation proves that our approach has achieved competitive performance on Wikipedia sentences from various domains.

2. DATA

Following the same data collection approach as in [22],

[1]http://stats.wikimedia.org/EN/Sitemap.htm
[2]http://wikipedia.org/wiki/Wikipedia:NPOV

WebSci '16, May 22 - 25, 2016, Hannover, Germany

ISBN 978-1-4503-4208-7/16/05...$15.00

DOI: http://dx.doi.org/10.1145/2908131.2908177

we built a dataset containing almost all the sentences in Wikipedia explicitly mentioning one of our targeted 219 named entities. To evaluate the classifier's performance on sentences from various domains, the named entities were selected evenly from four popular categories, which were: multinational corporations, politicians, celebrities and sport stars. The resulting dataset contained 1,196,403 sentences. We used CrowdFlower to annotate 5037 sentences (23 sentences per named entity) selected from the dataset, into two categories: *reputation-influential* sentence and *reputation non-influential* sentence.

Due to the NPOV policy of Wikipedia, most sentences in Wikipedia are impartial and narrative. This kind of sentences have minor influence on the mentioned named entities' reputation, as most words included in these sentences are neutral, non-judgmental and unbiased. To avoid the situation that *reputation non-influential* sentences dominate the dataset to be annotated, we applied a simple strategy to increase the percentage of sentences that carried strong subjective (i.e. weak objective, as these were complementary) words into the dataset to be annotated. First, for each named entity, we calculated the average objective score ($AvgObjScore$) of all the words in each sentence s that mentioned this named entity, as in Eq. 1.

$$AvgObjScore(s) = \frac{\sum_{i=1}^{m} ObjScore(w_i)}{m} \quad (1)$$

In Eq. 1, w_i is the i^{th} word in s that is included in SentiWordNet [3]; $ObjScore(w_i)$ is w_i's objective score in SentiWordNet; m is the total number of words in s that are included in SentiWordNet. Second, half of the sentences in the dataset to be annotated were sentences with the least $AvgObjScore$. This was due to the fact that words contained in these sentences were relatively strongly subjective in general, thus they were more likely to be *reputation-influential*, and promote empathy feelings of Wikipedia users. Third, the other half of the sentences in the dataset to be annotated were sentences randomly sampled from the rest, to alleviate the strong subjective polarisation of our dataset. Thus, the dataset to be annotated was a combination of the sentences with low $AvgObjScore$ and the sentences retrieved from random sampling.

We provided the annotators with the sentences to be annotated and their corresponding mentioned named entities, and asked the annotators to label these sentences, based on their judgement — if these sentences would influence the mentioned named entities' reputation. For the *reputation-influential* sentences, we asked the annotators to further response what kind of influence these sentences would have, *positive* or *negative*. There were three annotators allocated to pass judgment independently on each sentence, and more than 1,000 annotators with different backgrounds participated our task. The annotators were free to annotate any number of sentences. Crowdflower provided us the confidence score[3] of each response for each sentence, which was calculated as the agreement among multiple annotators on this response weighted by their accuracy on our test questions. For each sentence, the response with the highest confidence score was chosen as the annotation of the sentence. Similar to [13], we filtered out annotations with low confidence scores to improve the reliability. For our application, only annotations with confidence scores higher than 0.75 were applied to train the classifiers, which left us with 1,147 *reputation non-influential* sentences, 461 *positive reputation-influential* sentences and 228 *negative reputation-influential* sentences.

[3]http://success.crowdflower.com/hc/en-us/articles/201855939-How-to-Calculate-a-Confidence-Score

3. METHOD

Our goal is to detect the *positive reputation-influential* and *negative reputation-influential* sentences from Wikipedia. We cast the *reputation-influential* sentence detection as a cross-domain sentence multi-classification problem. All the sentences are classified into three categories: *positive reputation-influential* sentences, *negative reputation-influential* sentences and *reputation non-influential* sentences. Similar to [12, 4], we apply a 2-step binary classification method for multi-classification. In the first step, the sentences are classified into two categories: *reputation-influential* sentences and *reputation non-influential* sentences. The *reputation-influential* sentences are further classified into *positive reputation-influential* sentences and *negative reputation-influential* sentences. We selected for both steps a Support Vector Machine (SVM) classifier with RBF kernel, a most widely used classifier in sentence classification applications.

As, under the strong influence of the NPOV policy, the numbers of sentences from different categories in the annotated dataset is still quite unbalanced, we perform downsampling on the sentences from the *reputation non-influential* category to balance the number of *reputation-influential* sentences and *reputation non-influential* sentences.

It is hard for traditional fully-supervised approaches to achieve good performance in cross-domain scenarios, because they need a large number of annotated sentences from various domains. In our approach, we tackle this problem from the following directions: first, we prioritise domain independent features, when performing feature extraction; second, we leverage unlabelled sentences, to provide topical and word embedding features, in order to boost the performance of traditional classifiers; third, we incorporate many lexicons, to provide rich domain independent prior knowledge for classification.

Since it is difficult to clarify which features are useful for which step, we run various tests with various subsets of the full feature set for both steps, to select the features that were performing best. The results of this process are further presented in Table 1. To diminish the risk of introducing too many unrelevant features and reduce the dimensionality of the training matrix, we incorporate Randomized Logistic Regression [9], as a further feature selection step after fixing the feature set for one classifier. Next, we introduce the full feature set used.

3.1 Baseline features (FS1)

The first set to choose from are baseline features mostly used in classifiers for sentence classification, as follows:

Number of words: Number of words in the sentence.

N-gram features: The tf-idf values of unigrams and bigrams in the sentence.

Punctuation features: Number of question marks and number of exclamation marks in the sentence.

POS-tag features: We use the Stanford POS tagger [17] to POS-tag all sentences. Numbers of adjectives, adverbs, verbs and nouns are included into the feature set.

Dependency features: We represent all the dependen-

cies as features, to capture grammatical relationships between words in the sentence. This is achieved via the Stanford dependency parser [6]. For example, in the sentence "German Chancellor Angela Merkel and US Vice President Joe Biden condemned the attack on the US mission.", even trigrams are not able to capture the nominal subject relationship between words "Merkel" and "condemned". We represent this dependency as *nsubj_condemned_Merkel* and include the number of its occurrences into our feature set.

3.2 Lexicon features (FS2)

We have collected all the commonly used biased lexicons and sentiment lexicons, and have transferred the prior knowledge contained in these lexicons into features, as follows.

Opinion Lexicon features: The Opinion Lexicon [8] contains a positive opinion words list and a negative opinion words list. We include the numbers of positive and negative opinion words from the lexicon that occur in the sentence into the feature set.

Biased Lexicon features: The Biased Lexicon [14] contains a list of biased words. We include the number of biased words from the lexicon that occur in the sentence into the feature set.

MPQA Subjectivity Lexicon features: The MPQA Subjectivity Lexicon [19] contains a list of words, with each word's level of subjectivity(strongly subjective or weakly subjective), POS tag and prior polarity(positive, neutral or negative) provided. We lemmatise both the words in the lexicon and the words in the sentence, and include the number of strong and weak subjective words from the lexicon that occur in the sentence, as well as the number of positive, neutral and negative words occurring in the sentence into the feature set.

SentiWordNet Lexicon features: The SentiWordNet Lexicon [3] contains a list of words, with each word's POS tag, positive score ($PosScore$), negative score ($NegScore$) and objective score ($ObjScore$) provided, where $ObjScore = 1 - PosScore - NegScore$. We use w_i to denote the words from the lexicon that occur in the sentence. The following features derived based on SentiWordNet Lexicon are included into the feature set: (i) Number of w_i, denoted by m; (ii) Number of w_i with the $ObjScore$ higher than $PosScore$+$NegScore$; (iii) Number of w_i with the $PosScore$ higher than $NegScore$; (iv) Number of w_i with the $NegScore$ higher than $PosScore$; (v) The sum of $ObjScore$, $PosScore$ and $NegScore$ of w_i; (vi) The maximum of $ObjScore$, $PosScore$ and $NegScore$ of w_i; (vii) The average of $ObjScore$, $PosScore$ and $NegScore$ of w_i.

MSOL Lexicon features: The MSOL Lexicon [11] provides both single-word entries and multi-word expressions with their sentiment labels. We include the number of positive and negative single-word entries/multi-word expressions from the lexicon that occur in the sentence into the feature set.

3.3 Unsupervised features

As we have a large dataset with only a small part of it annotated, thus we propose to use unsupervised features, aiming at gaining additional knowledge from the whole dataset.

Latent Dirichlet Allocation (LDA) topic features (FS3): We train LDA models [5] with all the sentences in the original dataset, no matter if they are annotated or unannotated, with a wide different numbers of predefined topics $K = \{50, 100, 200, 300, 400, 500\}$. Then we represent each sentence with its topical distribution vector, with each dimension in the vector denoting the topic proportion for topic k. We incorporate the sentence's topical distribution representation vectors into the feature set, and test the classifier's performance with different K.

Word embedding features (FS4): In [10], researchers proposed the continuous Skip-gram model to learn word embedding representations in a new vector space $\mathcal{R}^N$, in order to capture syntactic and semantic word relationships. We train word2vec models [10] on all the sentences in the original dataset, using Gensim [15], with a wide range of vector space dimensionalities $N = \{50, 100, 200, 300, 400, 500\}$, in order to obtain the most suitable representation vectors for all the words occurring in the original dataset.

Word embedding features have been applied in sentence classification tasks, such as [18]. Unlike [18], when generating the sentence-level embedding representation vectors, we use tf-idf values to weigh each word, in order to decrease the influence of unimportant words. We use $\vec{v}(w_i) \in \mathcal{R}^N$ to denote the embedding representation vector of word w_i in the sentence, and $tfidf(w_i)$ to denote the tf-idf value of w_i in the original dataset. The embedding representation vector of sentence s can be calculated as:

$$\vec{v}(s) = \frac{\sum_{i=1}^{m} tfidf(w_i) \cdot \vec{v}(w_i)}{m}, \quad (2)$$

where m denotes the total number of words in the sentence, and $\vec{v}(s) \in \mathcal{R}^N$.

The embedding representation vector of the sentence is included into our feature set.

4. RESULTS

We have investigated two application scenarios and we focused on the average F1 scores achieved in different scenarios. The first scenario was binary classification, in which we only aimed at detecting *reputation-influential* sentences. The second scenario was multi-classification, in which we aimed at deciding both whether one sentence was *reputation-influential* and the direction in which it influenced the entity's reputation.

4.1 Reputation-influential Sentence Detection

We performed feature selection manually by analysing the classifier's performance with different feature sets on the basis of Randomized Logistic Regression, using 10-fold cross-validation. We did not totally rely on Randomized Logistic Regression for feature selection, in order to discover the most effective kind of features, and discard redundant features. For different feature sets, we used grid search to choose the most suitable number of topics for the LDA-based topical features K, the dimensionality of the word embedding vector representation N, the penalty parameter of the SVM classifier C, the kernel coefficient γ.

In Table 1, we use FS1 to denote baseline features, FS2 to denote lexicon features, FS3 to denote topical features and FS4 to denote word embedding features. FS1234 represents the combination of FS1, FS2, FS3 and FS4. We use P to represent precision, R to represent recall and F1 to represent F1 score. From Table 1, we can see that the classifier using lexicon features, topical features and word embedding features (FS234) achieves the best performance, which outperforms the benchmark classifier just using base-

Table 1: Performance of classifiers with different feature sets

	Influential			Non-influential		
Features	P	R	F1	P	R	F1
FS1	0.817	0.386	0.521	0.606	0.916	0.729
FS12	0.745	0.750	0.747	0.755	0.747	0.750
FS123	0.765	0.777	0.771	0.780	0.766	0.773
FS124	0.760	0.781	0.769	0.781	0.758	0.768
FS134	0.768	0.711	0.737	0.738	0.791	0.763
FS34	0.771	0.717	0.743	0.743	0.792	0.766
FS24	0.790	0.770	0.780	0.782	0.799	0.790
FS23	0.711	0.726	0.718	0.728	0.711	0.719
FS234	0.781	0.795	**0.788**	0.807	0.782	**0.795**
FS1234	0.788	0.783	0.786	0.786	0.790	0.783

line features (FS1). The best performance is achieved with FS234 when $K = 100$, $N = 100$, $C = 1$ and $\gamma = 0.005$. We find that the increase in the number of topics and the dimensionality of the word embedding vector representation do not always lead to an improvement of the classifier's performance. This is as a larger feature spaces is less able to generalise for sentences from various domains.

Both lexicon features and unsupervised features help to increase the average F1 score. The most helpful features are the word embedding features. This illustrates that word embedding features are the best semantic generalisations of the original Wikipedia sentences from various domains. The average F1 score drops after adding baseline features on the basis of lexicon features, topical features and word embedding features. This is because most baseline features, such as n-grams or dependency features, are domain dependent, and the classifier is experiencing the overfitting problem. On one hand, the lexicon features, topical features and word embedding features already capture any useful patterns in baseline features. On the other hand, the baseline features include some irrelevant and redundant features that can hurt the classifier's performance. These factors allow the classifier which excludes the baseline features to outperform other classifiers, including the one with all available features.

4.2 Positive Reputation-influential, Reputation Non-influential and Negative Reputation-influential sentences

We conducted similar experiments as in 4.1 to select the best feature sets and hyperparameters for the classifier used to distinguish between *positive reputation-influential* sentences and *negative reputation-influential* sentences, and the classifier for one-vs-one multi-classification. Interestingly, the best feature sets for these two classifiers were also FS234. We compared our two-step binary classification method with the benchmark one-vs-one multi-classification method [7]. Table 2 shows the performance comparison of these two methods. An average F1 score of 0.717 is achieved with our two-step binary classification method when classifying all the Wikipedia sentences into three categories, higher than the average F1 score of the baseline one-vs-one multi-classification method, which is 0.705. This is because the *positive reputation-influential* and *negative reputation-influential* sentences share some common features, thus the combination of the sentences of these two categories provides the classifier more information than differentiating sentences of these two categories from the sentences of the*reputation non-influential* category separately.

5. RELATED WORK

Various features have been considered when tackling the sentence classification problem. For example, n-grams [4, 1], POS tags [4, 1], lexicon-based features [4, 2], dependency features [2], LDA-based topical features [20] and word embedding features [16]. This paper defined and tackled a novel sentence classification problem: detecting *reputation-influential* sentences from encyclopaedic content. From [16], we learnt that a classifier with combined hand-crafted features and word embedding features can outperform several baseline approaches. To our best knowledge, our classifiers jointly considered all the available state-of-the-art features, and are different from former researches in the way of extracting and applying them, such as the SentiWordNet features and the word embedding features. Our approach has achieved a promising performance for our task.

Another relevant track of research is Wikipedia-related text mining. Rather than focusing on the main content of Wikipedia, [14] trained linguistic models for detecting biased language on Wikipedia's historical edits, and they achieved 58.70% accuracy; [21] explored the sentiment bias of multilingual Wikipedia on events at article level; [22] compared the sentiment bias of multilingual Wikipedia on entities at corpus level. The work can be seen as locating the sentences that influence the reputation of entities, which further leads to the sentiment bias detected in [22, 21].

6. CONCLUSION

In this paper, we have proposed an approach to detect *reputation-influential* sentences in Wikipedia. We have applied several lexicons, to generate domain independent lexicon features, and have leveraged an unlabelled dataset, to generate topical features and word embedding features. All these features have been proven to be functional in our experiments. Our classifier can achieve an average F1 score of 0.792 for cross-domain binary sentence classification. We have adopted a two-step binary classification method when performing the task of classifying all the Wikipedia sentences into three categories: *positive reputation-influential*, *reputation non-influential* and *negative reputation-influential*. This method outperformed a benchmark one-vs-one multi-classification method and reached an average F1 score of 0.717. The detected *positive reputation-influential* sentences and *negative reputation-influential* sentences are the sentences that Wikipedia users are potentially most interested in, thus the user experience could be improved by highlighting them; alternatively, they could also help the administrators to better apply the NPOV policy of Wikipedia. Although we have limited our application scenario to *reputation-influential* sentences detection on Wikipedia, the proposed features and multi-classification method could also be helpful for other sentence classification tasks.

7. REFERENCES

[1] A. Abbasi, H. Chen, and A. Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12, 2008.

Table 2: Performance comparison between two-step binary classification and one-vs-one multi-classification

	Positive influential			**Non-influential**			**Negative influential**			**Avg.**
TYPE	P	R	F1	P	R	F1	P	R	F1	F1
Multi	0.723	0.695	0.708	0.672	0.684	**0.677**	0.725	0.733	0.729	**0.705**
Two-step	0.715	0.713	**0.714**	0.668	0.673	0.670	0.766	0.768	**0.767**	**0.717**

[2] A. Athar. Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL 2011 student session*, pages 81–87. Association for Computational Linguistics, 2011.

[3] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[4] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics, 2010.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[6] D. Chen and C. D. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750, 2014.

[7] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011.

[8] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[9] N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] S. Mohammad, C. Dunne, and B. Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 599–608. Association for Computational Linguistics, 2009.

[12] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[13] A. Ramesh, S. H. Kumar, J. Foulds, and L. Getoor. Weakly supervised models of aspect-sentiment for online course discussion forums. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[14] M. Recasens, C. Danescu-Niculescu-Mizil, and D. Jurafsky. Linguistic models for analyzing and detecting biased language. In *ACL (1)*, pages 1650–1659, 2013.

[15] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[16] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1555–1565, 2014.

[17] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[18] R. Townsend, A. Tsakalidis, Y. Zhou, B. Wang, M. Liakata, A. Zubiaga, A. Cristea, and R. Procter. Warwickdcs: From phrase-based to target-specific sentiment recognition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 657–663, Denver, Colorado, June 2015. Association for Computational Linguistics.

[19] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[20] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984. ACM, 2012.

[21] Y. Zhou, A. I. Cristea, and Z. Roberts. Is wikipedia really neutral? A sentiment perspective study of war-related wikipedia articles since 1945. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, PACLIC 29, Shanghai, China, October 30 - November 1, 2015*, 2015.

[22] Y. Zhou, E. Demidova, and A. I. Cristea. Who likes me more? analysing entity-centric language-specific bias in multilingual wikipedia. In *Proceedings of the 31th Annual ACM Symposium on Applied Computing*, SAC '16, 2016.