QCRI: Answer Selection for Community Question Answering - Experiments for Arabic and English

QCRI: Answer Selection for Community Question Answering – Experiments for Arabic and English

Massimo Nicosia[1], **Simone Filice**[2], **Alberto Barrón-Cedeño**[2], **Iman Saleh**[3], **Hamdy Mubarak**[2], **Wei Gao**[2], **Preslav Nakov**[2], **Giovanni Da San Martino**[2], **Alessandro Moschitti**[2], **Kareem Darwish**[2], **Lluís Màrquez**[2], **Shafiq Joty**[2] and **Walid Magdy**[2]

[1] University of Trento [2] Qatar Computing Research Institute [3] Cairo University

massimo.nicosia@unitn.it

{sfilice,albarron,hmubarak,wgao,pnakov,gmartino}@qf.org.qa

{amoschitti,kdarwish,lmarquez,sjoty,wmagdy}@qf.org.qa

iman.saleh@fci-cu.edu.eg

Abstract

This paper describes QCRI's participation in SemEval-2015 Task 3 "Answer Selection in Community Question Answering", which targeted real-life Web forums, and was offered in both Arabic and English. We apply a supervised machine learning approach considering a manifold of features including among others word n-grams, text similarity, sentiment analysis, the presence of specific words, and the context of a comment. Our approach was the best performing one in the Arabic subtask and the third best in the two English subtasks.

1 Introduction

SemEval-2015 Task 3 "Answer Selection in Community Question Answering" challenged the participants to automatically predict the appropriateness of the answers in a community question answering setting (Màrquez et al., 2015). Given a question $q \in Q$ asked by user u_q and a set of comments C, the main task was to determine whether a comment $c \in C$ offered a suitable answer to q or not.

In the case of Arabic, the questions were extracted from *Fatwa*, a community question answering website about Islam.[1] Each question includes five comments, provided by scholars on the topic, each of which has to be automatically labeled as (*i*) `DIRECT`: a direct answer to the question; (*ii*) `RELATED`: not a direct answer to the question but with information related to the topic; and (*iii*) `IRRELEVANT`: an answer to another question, not related to the topic. This is subtask A, Arabic.

In the case of English, the dataset was extracted from *Qatar Living*, a forum for people to pose questions on multiple aspects of daily life in Qatar.[2] Unlike *Fatwa*, the questions and comments in this dataset come from regular users, making them significantly more varied, informal, open, and noisy. In this case, the input to the system consists of a question and a variable number of comments, each of which is to be labeled as (*i*) `GOOD`: the comment is definitively relevant; (*ii*) `POTENTIAL`: the comment is potentially useful; and (*iii*) `BAD`: the comment is irrelevant (e.g., it is part of a dialogue, unrelated to the topic, or it is written in a language other than English). This is subtask A, English.

Additionally, a subset of the questions required a `YES`/`NO` answer, and there was another subtask for them, which asked to determine whether the overall answer to the question, according to the evidence provided by the comments, is (*i*) `YES`, (*ii*) `NO`, or (*iii*) `UNSURE`. This is subtask B, English.

Details about the subtasks and the experimental settings can be found in (Màrquez et al., 2015).

Below we describe the supervised learning approach of QCRI, which considers different kinds of features: lexical, syntactic and semantic similarities; the context in which a comment appears; n-grams occurrence; and some heuristics. We ranked first in the Arabic, and third in the two English subtasks.

The rest of the paper is organized as follows: Section 2 describes the features used, Section 3 discusses our models and our official results, and Section 4 presents post-competition experiments and offers some final remarks.

[1] `http://fatwa.islamweb.net`

[2] `http://www.qatarliving.com/forum`

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 203–209,
Denver, Colorado, June 4-5, 2015.

2 Features

In this section, we describe the different features we considered including similarity measures (Section 2.1), the context in which a comment appears (Section 2.2), and the occurrence of certain vocabulary and phrase triggers (Sections 2.3 and 2.4). How and where we apply them is discussed in Section 3. Note that while our general approach is based on supervised machine learning, some of our contrastive submissions are rule-based.

2.1 Similarity Measures

The similarity features measure the similarity $sim(q, c)$ between the question and a target comment, assuming that high similarity signals a `GOOD` answer. We consider three kinds of similarity measures, which we describe below.

2.1.1 Lexical Similarity

We compute the similarity between word n-gram representations ($n = [1, \ldots, 4]$) of q and c, using the following lexical similarity measures (after stopword removal): greedy string tiling (Wise, 1996), longest common subsequences (Allison and Dix, 1986), Jaccard coefficient (Jaccard, 1901), word containment (Lyon et al., 2001), and cosine similarity. We further compute cosine on lemmata and POS tags, either including stopwords or not.

We also use similarity measures, which weigh the terms using the following three formulæ:

$$sim(q, c) = \sum_{t \in q \cap c} idf(t) \quad (1)$$

$$sim(q, c) = \sum_{t \in q \cap c} log(idf(t)) \quad (2)$$

$$sim(q, c) = \sum_{t \in q \cap c} log\left(1 + \frac{|C|}{tf(t)}\right) \quad (3)$$

where $idf(t)$ is the inverse document frequency (Sparck Jones, 1972) of term t in the entire Qatar Living dataset, C is the number of comments in this collection, and $tf(t)$ is the term frequency of the term in the comment. Equations 2 and 3 are variations of idf; cf. Nallapati (2004).

For subtask B, we further considered the cosine similarity between the tf-idf-weighted intersection of the words in q and c.

2.1.2 Syntactic Similarity

We further use a partial tree kernel (Moschitti, 2006) to calculate the similarity between the question and the comment based on their corresponding shallow syntactic trees. These trees have word lemmata as leaves, then there is a POS tag node parent for each lemma leaf, and POS tag nodes are in turn grouped under shallow parsing chunks, which are linked to a root sentence node; finally, all root sentence nodes are linked to a super root for all sentences in the question/comment.

2.1.3 Semantic Similarity

We apply three approaches to build word-embedding vector representations, using (*i*) latent semantic analysis (Croce and Previtali, 2010), trained on the Qatar Living corpus with a word co-occurrence window of size ± 3 and producing a vector of 250 dimensions with SVD (we produced a vector for each noun in the vocabulary); (*ii*) GloVe (Pennington et al., 2014), using a model pre-trained on *Common Crawl (42B tokens)*, with 300 dimensions; and (*iii*) COMPOSES (Baroni et al., 2014), using previously-estimated predict vectors of 400 dimensions.[3] We represent both q and c as a sum of the vectors corresponding to the words within them (neglecting the subject of c). We compute the cosine similarity to estimate $sim(q, c)$.

We also experimented with *word2vec* (Mikolov et al., 2013) vectors pre-trained with both *cbow* and *skipgram* on news data, and also with both word2vec and GloVe vectors trained on Qatar Living data, but we discarded them as they did not help us on top of all other features we had.

2.2 Context

Comments are organized sequentially according to the time line of the comment thread. Whether a question includes further comments by the person who asked the original question or just several comments by the same user, or whether it belongs to a category in which a given kind of answer is expected, are all important factors. Therefore, we consider a set of features that try to describe a comment in the context of the entire comment thread.

[3] They are available at `http://nlp.stanford.edu/projects/glove/` and `http://clic.cimec.unitn.it/composes/semantic-vectors.html`

We have boolean context features that explore the following situations:

- c is written by u_q (i.e., the same user behind q),
- c is written by u_q and contains an acknowledgment (e.g., *thank**, *appreciat**),
- c is written by u_q and includes further question(s), and
- c is written by u_q and includes no acknowledgments nor further questions.

We further have numerical features exploring whether comment c appears in the proximity of a comment by u_q; the assumption is that an acknowledgment or further questions by u_q could signal a bad answer:

- among the comments following c there is one by u_q containing an acknowledgment,
- among the comments following c there is one by u_q not containing an acknowledgment,
- among the comments following c there is one by u_q containing a question, and
- among the comments preceding c there is one by u_q containing a question.

The numerical value of these last four features is determined by the distance k, in number of comments, between c and the closest comment by u_q ($k = \infty$ if no comments by u_q exist):

$$f(c) = \max\left(0,\ \ 1.1 - (k \cdot 0.1)\right) \qquad (4)$$

We also tried to model potential dialogues by identifying interlacing comments between two users. Our dialogue features rely on identifying conversation chains between two users:

$$u_i \rightarrow \ldots \rightarrow u_j \rightarrow \ldots \rightarrow u_i \rightarrow \ldots \rightarrow [u_j]$$

Note that comments by other users can appear in between the nodes of this "pseudo-conversation" chain. We consider three features: whether a comment is at the beginning, in the middle, or at the end of such a chain. We have copies of these three features for the special case when $u_q = u_j$.

We are also interested in modeling whether a user u_i has been particularly active in a question thread. Thus, we add one boolean feature: whether u_i wrote more than one comment in the current thread.

Three more features identify the first, the middle and the last comments by u_i. One extra feature counts the total number of comments written by u_i. Moreover, we empirically observed that the likelihood of a comment being `GOOD` decreases with its position in the thread. Therefore, we also include another real-valued feature: $\max(20, i)/20$, where i represents the position of the comment in the thread.

Finally, Qatar Living includes twenty-six different categories in which one could request information and advice. Some of them tend to include more open-ended questions and even invite discussion on ambiguous topics, e.g., *Socialising*, *Life in Qatar*, *Qatari Culture*. Some other require more precise answers and allow for less discussion, e.g., *Visas and Permits*. Therefore, we include one boolean feature per category to consider this information.

2.3 Word n-Grams

Our features include n-grams, independently obtained from both the question and the comment: $[1, 2]$-grams for Arabic, and stopworded $[1, 2, 3]$-grams for English. That is, each n-gram appearing in the texts becomes a member of the feature vector. The value for such features is tf-idf, with idf computed on the entire Qatar Living dataset.

Our aim is to capture the words that are associated with questions and comments in the different classes. We assume that objective and clear questions would tend to produce objective and `GOOD` comments. On the other hand, subjective or badly formulated questions would call for `BAD` comments or discussion, i.e., dialogues, among the users. This can be reflected by the vocabulary used, regardless of the topic of the formulated question. This is also true for comments: the occurrence of particular words could make a comment more likely to be `GOOD` or `BAD` , regardless of what question was asked.

2.4 Heuristics

Exploring the training data, we noticed that many `GOOD` comments suggested visiting a Web site or contained an email address. Therefore, we included two boolean features to verify the presence of URLs or emails in c. Another feature captures the length of c, as longer (`GOOD`) comments usually contain detailed information to answer a question.

2.5 Polarity

These features, which we used for subtask B only, try to determine whether a comment is positive or negative, which could be associated with YES or NO answers. The polarity of a comment c is

$$pol(c) = \sum_{w \in c} pol(w) \quad (5)$$

where $pol(w)$ is the polarity of word w in the NRC Hashtag Sentiment Lexicon v0.1 (Mohammad et al., 2013). We disregarded $pol(w)$ if its absolute value was less than 1.

We further use boolean features that check the existence of some keywords in the comment. Their values are set to true if c contains words like (*i*) *yes*, *can*, *sure*, *wish*, *would*, or (*ii*) *no*, *not*, *neither*.

2.6 User Profile

With this set of features, we aim to model the behavior of the different participants in previous queries. Given comment c by user u, we consider the number of GOOD, BAD, POTENTIAL, and DIALOGUE comments u has produced before.[4] We also consider the average word length of GOOD, BAD, POTENTIAL, and DIALOGUE comments. These features are computed both considering all questions and taking into account only those from the target category. [5]

3 Submissions and Results

Below we describe our primary submissions for the three subtasks; then we discuss our contrastive submissions. Our classifications for subtask A, for both Arabic and English, are at the comment level. Table 1 shows our official results at the competition; all reported F_1 values are macro-averaged.

3.1 Primary Submissions

Arabic. We used logistic regression. The features are lexical similarities (Section 2.1) and n-grams (Section 2.3). In a sort of stacking, the output of our cont_1 submission is included as another feature (cf. Section 3.2).

This submission achieved the first position in the competition ($F_1 = 78.55$, compared to 70.99 for the second one). It showed a particularly high performance when labeling RELATED comments.

English, subtask A. Here we used a linear SVM, and a one-vs.-rest approach as we have a multiclass problem. The features for this submission consist of lexical, syntactic, and semantic similarities (Section 2.1), context information (Section 2.2), n-grams (Section 2.3), and heuristics (Section 2.4). Similarly to Arabic, the output of our rule-based system from the cont_2 submission is another feature.

This submission achieved the third position in the competition ($F_1 = 53.74$, compared to 57.19 for the top one). POTENTIAL comments proved to be the hardest, as the border with respect to the rest of the comments is very fuzzy. Indeed, a manual inspection on some random comments has shown that distinguishing between GOOD and POTENTIAL comments is often impossible.

English, subtask B. Following the organizers' manual labeling strategy for the YES/NO questions (Màrquez et al., 2015), we used three steps: (*i*) identifying the GOOD comments for q; (*ii*) classifying each of them as YES, NO, or UNSURE; and (*iii*) aggregating these predictions to the question level (majority). In case of a draw, we labeled the question as UNSURE.[6]

Step (*i*) is subtask A. For step (*ii*), we train a classifier as for subtask A, including the polarity and the user profile features (cf. Sections 2.5 and 2.6).[7]

This submission achieved the third position in the competition: $F_1 = 53.60$, compared to 63.70 for the top one. Unlike the other subtasks, for which we trained on both the training and the testing datasets, here we used the training data only, which was due to instability of the results when adding the development data. Post-submission experiments revealed this was due to some bugs as well as to unreliability of some of the statistics. Further discussion on this can be found in Section 4.3.

[4] About 72% of the comments in the test set were written by users who had been seen in the training/development set.

[5] In Section 4.3, we will observe that computing these category-level features was not a good idea.

[6] The majority class in the training and dev. sets (YES) could be the default answer. Still, we opted for a conservative decision: choosing UNSURE if no enough evidence was found.

[7] Even if the user profile information seems to fit for subtask A rather than B, at development time it was effective for B only.

ar	DIRECT	IRREL	RELATED	F_1
primary	77.31	91.21	67.13	78.55
$cont_1$	74.89	91.23	63.68	76.60
$cont_2$	76.63	90.30	63.98	76.97
en A	GOOD	BAD	POT	F_1
primary	78.45	72.39	10.40	53.74
$cont_1$	76.08	75.68	17.44	56.40
$cont_2$	75.46	72.48	7.97	51.97
en B	YES	NO	UNSURE	F_1
primary	80.00	44.44	36.36	53.60
$cont_1$	75.68	0.00	0.00	25.23
$cont_2$	66.67	33.33	47.06	49.02

Table 1: Per-class and macro-averaged F_1 scores for our official primary and contrastive submissions to SemEval-2015 Task 3 for Arabic (ar) and English (en), subtasks A and B.

3.2 Contrastive Submissions

Arabic. We approach our contrastive submission 1 as a ranking problem. After stopword removal and stemming, we compute $sim(q, c)$ as follows:

$$sim(q, c) = \frac{1}{|q|} \sum_{t \in q \cap c} \omega(t) \quad (6)$$

where we empirically set $\omega(t) = 1$ if t is a 1-gram, and $\omega(t) = 4$ if t is a 2-gram. Given the 5 comments $c_1, \ldots, c_5 \in C$ associated with q, we map the maximum similarity $\max_C sim(q, c)$ to a maximum 100% similarity and we map the rest of the scores proportionally. Each comment is assigned a class according to the following ranges: [80, 100]% for DIRECT, (20,80)% for RELATED, and [0,20]% for IRRELEVANT. We manually tuned these threshold values on the training data.

As for the contrastive submission 2, we built a binary classifier DIRECT vs. NO-DIRECT using logistic regression. We then sorted the comments according to the classifier's prediction confidence and we assigned labels as follows: DIRECT for the top ranked, RELATED for the second ranked, and IRRELEVANT for the rest. We only included lexical similarities as features, discarding those weighted with idf variants.

The performance of these two contrastive submissions was below but close to that of our primary submission (F_1 of 76.60 and 76.97, vs. 78.55 for primary), particularly for IRRELEVANT comments.

English, subtask A. Our contrastive submission 1, uses the same features and schema as our primary submission, but with SVM^{light} (Joachims, 1999), which allows us to deal with the class imbalance by tuning the j parameter, i.e., the cost of making mistakes on positive examples. This time, we set the C hyper-parameter to the default value. As we focused on improving the performance on POTENTIAL instances, we obtained better results for this category (F_1 of 17.44 vs. 10.40 for POTENTIAL), surpassing the overall performance for our primary submission (F_1 of 56.40 vs. 53.74).

Our contrastive submission 2 is similar to our Arabic contrastive submission 1, using the same ranges, but now for GOOD, POTENTIAL, and BAD. We also have post-processing heuristics: c is classified as GOOD if it includes a URL, starts with an imperative verb (e.g., *try*, *view*, *contact*, *check*), or contains *yes words* (e.g., *yes*, *yep*, *yup*) or *no words* (e.g., *no*, *nooo*, *nope*). Moreover, comments written by the author of the question or including acknowledgments are considered dialogues, and thus classified as BAD. The result of this submission is slightly lower than for primary and contrastive 1: F_1=51.97.

English, subtask B. Our contrastive submission 1 is like our primary, but is trained on both the training and the development data. The reason for the low results (an F_1 of 25.23, compared to 53.60 for the primary) were bugs in the polarity features (cf. Section 2.5) and lack of statistics for properly estimating the category-level user profiles (cf. Section 2.6).

The contrastive submission 2 is a rule-based system. A question is answered as YES if it starts with affirmative words: *yes*, *yep*, *yeah*, etc. It is labeled as NO if it starts with negative words: *no*, *nop*, *nope*, etc. The answer to q becomes that of the majority of the comments: UNSURE in case of tie. It is worth noting the comparably high performance when dealing with UNSURE questions: F_1=47.06, compared to 36.36 for our primary submission.

4 Post-Submission Experiments

We carried out post-submission experiments in order to understand how different feature families contributed to the performance of our classifiers; the results are shown in Table 2. We also managed to improve our performance for all three subtasks.

ar (only)	DIR	IRREL	REL	F_1
n-grams	30.40	41.07	72.27	47.91
$cont_1$	74.89	63.68	91.23	76.60
similarities	61.83	25.63	82.55	56.67
ar (without)	DIR	REL	IRREL	F_1
n-grams	75.51	91.31	63.85	76.89
$cont_1$	69.50	82.85	50.87	67.74
similarities	77.24	91.07	67.76	**78.69**
en A (only)	GOOD	BAD	POT	F_1
context	67.65	45.03	11.51	47.90
n-grams	71.22	40.12	5.99	44.86
heuristics	76.46	41.94	7.11	52.57
similarities	62.93	44.58	9.62	46.16
lexical	62.25	41.46	8.66	44.82
syntactic	59.18	36.20	0.00	36.47
semantic	55.56	40.42	9.92	42.16
en A (without)	GOOD	BAD	POT	F_1
context	76.05	41.53	8.98	51.50
n-grams	77.25	45.56	12.23	**55.17**
heuristics	73.84	65.33	6.81	48.66
similarities	78.02	71.82	9.88	53.24
lexical	78.23	72.81	9.91	53.65
syntactic	78.81	43.89	9.91	53.73
semantic	78.41	71.82	10.30	53.51
en B	YES	NO	UNS	F_1
$post_1$	78.79	57.14	20.00	51.98
$post_2$	85.71	57.14	25.00	**55.95**
primary	D/G/Y	I/B/N	R/P/U	F_1
ar	77.31	91.21	67.13	78.55
en A	78.45	72.39	10.40	53.74
en B	80.00	44.44	36.36	53.60

Table 2: Post-submission results for Arabic (ar) and English (en), for subtasks A and B. The lines marked with *only* show results using a particular type of features only, while those marked as *without* show results when using all features but those of a particular type. The best results for each subtask are marked in bold; the results for our official primary submissions are included for comparison.

4.1 Arabic

We ran experiments with the same framework as in our primary submission by considering the subsets of features in isolation (*only*) or all features except for a subset (*without*). The n-gram features together with our $cont_1$ submission (recall that we also use $cont_1$ as a feature in our primary submission) allow for a slightly better performance than our —already winning— primary submission ($F_1 = 78.69$, compared to $F_1 = 78.55$). The $cont_1$ feature turns out to be the most important one, and, as it already contains similarity, combining it with other similarity features does not yield any further improvements.

4.2 English, Subtask A

We performed experiments similar to those we did for Arabic. According to the *only* figures, the heuristic features seem to be the most useful ones, followed by the context-based ones. The latter explore a dimension ignored by the rest: these features are completely uncorrelated and provide a good performance boost (as the *without* experiments show). On the other hand, using all features but the n-grams improves over the performance of our primary run ($F_1 = 55.17$ compared to $F_1 = 53.74$). This is an interesting but not very significant result as these features had already boosted our performance at development time. Further research is necessary.

4.3 English, Subtask B

Our post-submission efforts focused on investigating why learning from the training data only was considerably better than learning from training+dev. The output labels on the test set in the two learning scenarios showed considerable differences: when learning from training+dev, the predicted labels were YES for all but three cases. After correcting a bug in our implementation of the polarity-related features, the result when learning on training+dev became F_1=51.98 (Table 2, $post_1$). Further analysis showed that the features counting the number of GOOD, BAD, and POTENTIAL comments within categories by the same user (cf. Section 2.6) varied greatly when computed on training and on training+dev, as the number of comments by a user in a category was, in most cases, too small to yield very reliable statistics. After discarding these three features, the F_1 raised to 55.95 (Table 2, $post_2$), which is higher than what we obtained at submission time. Note that, once again, the UNSURE class is by far the hardest to identify properly.

Surprisingly, learning with the bug-free implementation from the training set yielded a much higher F_1 of 69.35 on the test dataset (not shown in the table). Analysis revealed that the difference in performance was due to misclassifying just four questions. Indeed, the differences seem to occur due to the natural randomness of the classifier on a small test dataset and they cannot be considered statistically significant (Màrquez et al., 2015).

5 Conclusions and Future Work

We have presented the system developed by the team of the Qatar Computing Research Institute (QCRI) for participating in SemEval-2015 Task 3 on Answer Selection in Community Question Answering. We used a supervised machine learning approach and a manifold of features including word n-grams, text similarity, sentiment dictionaries, the presence of specific words, the context of a comment, some heuristics, etc. Our approach was the best performing one in the Arabic task, and the third best in the two English tasks.

We further presented a detailed study of which kinds of features helped most for each language and for each subtask, which should help researchers focus their efforts in the future.

In future work, we plan to use richer linguistic annotations, more complex kernels, and large semantic resources.

Acknowledgments

This research is developed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), Qatar Foundation in collaboration with MIT. It is part of the Interactive sYstems for Answer Search (Iyas) project.

References

Lloyd Allison and Trevor Dix. 1986. A bit-string longest-common-subsequence algorithm. *Inf. Process. Lett.*, 23(6):305–310, December.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '14, pages 238–247, Baltimore, MD, USA.

Danilo Croce and Daniele Previtali. 2010. Manifold Learning for the Semi-Supervised Induction of FrameNet Predicates: An Empirical Investigation. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '10, pages 7–16, Uppsala, Sweden.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

Thorsten Joachims. 1999. Making Large-scale Support Vector Machine Learning Practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods*, pages 169–184. MIT Press, Cambridge, MA, USA.

Caroline Lyon, James Malcolm, and Bob Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, EMNLP '01, pages 118–125, Pittsburgh, PA, USA.

Lluís Màrquez, James Glass, Walid Magdy, Alessandro Moschitti, Preslav Nakov, and Bilal Randeree. 2015. SemEval-2015 Task 3: Answer Selection in Community Question Answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, Denver, CO, USA.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, pages 746–751, Atlanta, GA, USA.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, pages 321–327, Atlanta, GA, USA.

Alessandro Moschitti. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 318–329. Springer Berlin Heidelberg.

Ramesh Nallapati. 2004. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 64–71, Sheffield, United Kingdom.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1532–1543, Doha, Qatar.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Michael Wise. 1996. Yap3: Improved detection of similarities in computer program and other texts. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '96, pages 130–134, New York, NY, USA.