QCRI: Experiments in Community Question Answering Selection in Arabic and English

Massimo Nicosia[1] and **Simone Filice**[2] and **Alberto Barrón-Cedeño**[2] and
Iman Saleh[3] and **Hamdy Mubarak**[2] and **Wei Gao**[2] and **Preslav Nakov**[2] and
Alessandro Moschitti[2] and **Giovanni Da San Martino**[2] and **Kareem Darwish**[2]
Lluís Màrquez[2] and **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,amoschitti}@qf.org.qa
{gmartino,kdarwish,lmarquez,sjoty,wmagdy}@qf.org.qa
iman.saleh@fci-cu.edu.eg

Abstract

6 pages + 2 for references

This paper describes the QCRI partitipation to SemEval-2015 Task 3 —Answer Selection in Community Question Answering— on both Arabic and English real-life question forums. We apply a supervised machine learning approach considering a manifold of features including word n-grams, text similarity, vocabulary polarity, and the presence of specific words, as well as the context of a comment, among others.

Our approach allowed us to get the first position in the Arabic task and the third position in the English task.

1 Introduction

The SemEval-2015 Task 3 —Answer Selection in Community Question Answering—, challenged participants in the problem of automatically identifying the appropriateness of user-generated answers in a community question answering setting both in Arabic and English (Màrquez et al., 2015). A question $q \in Q$, asked by user u_q, together with a set of comments C are given and the system is intended to determine whether a comment $c \in C$ offers a suitable answer to q or not.

In the case of Arabic, the questions were extracted from *Fatwa*, a community question answering website on the Islamic religion.[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.

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 are 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). We refer to this task as English task A. Additionally, a subset of the questions in the corpus requires a YES/NO answer. In this case the task is determining whether the overall answer to the question, according to the evidence provided within the comments, is (*i*) YES, (*ii*) NO, or if no evidence enough exists to make a decision, (*iii*) UNSURE. We refer to this as English task B.

In this paper we describe the supervised machine learning approach of QCRI. We considered different kinds of features, including lexical, syntactic and semantic similarities, the context in which a comment appears (e.g., before a comment where the person asking the question acknowledges), n-grams occurrence, and some heuristics on specific keywords. Our approach ranked 1st out of four teams in the

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

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

Arabic task, 3rd out of twelve in English Task A, and 3rd out of eight in English Task B.

The rest of the contribution is distributed as follows. Section 2 describes the features used in our approaches. Section 3 describes our prediction models and discuss the results obtained at competition time. Section 4 discusses some further post-competition experiments and offers some final remarks.

2 Features Description

Most of our approaches are built on top of supervised machine learning, whereas a few contrastive sbmissions were based on rule-based approaches. In this section we describe all the different features we considered including similarities (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 they are applied is discussed in Section 3.

2.1 Similarities

Our intuition is that the higher the similarity $sim(q, c)$, the higher the likelihood that c is a `GOOD` answer. We consider different types.

2.1.1 Lexical similarities

After stopwording, we compute $sim(q, c)$ for word n-gram representations of q and c ($n = [1, \ldots, 4]$), and different sim functions: 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 (cosine is also computed on lemmas and POS tags, either including stopwords or not).

Three other similarities are computed, weighting the terms by means of three formulæ:

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

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

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

where $idf(t)$ represents the inverse document frequency (Sparck Jones, 1972) of term t in the entire Qatar Living dataset, C represents the amount of comments in the entire collection, and $tf(t)$ represents the term frequency of the term in the comment. Equations 2 and 3 are variations of the IDF concept by Nallapati (2004).

Yet another similarity variation is considered (only for task B): the cosine similarity between the tf-idf-weighted vocabulary intersection of q and c.

2.1.2 Syntactic similarity Massimo; contrast?

Partial tree kernel (PTK) similarity between question and comment according to (Moschitti, 2006). The similarity is computed between shallow tree representations of q and c. Such trees have lemmas as leaves, each leaf has a parent node representing a part-of-speech tag, and part-of-speech nodes are grouped by chunks at the top level.

2.1.3 Semantic similarities

We apply three approaches to build word-embedding vector representations: (*i*) an instance of latent semantic analysis (Croce and Previtali, 2010), trained on the Qatar Living corpus applying a co-occurrence window of size ± 3 and coming out with a vector of dimension 250, after SVD reduction (we included an instance on the entire vocabulary and nouns only); (*ii*) GloVe (Pennington et al., 2014), using the pre-trained model *Common Crawl (42B tokens)*, with 300 dimensions;[3] and (*iii*) COMPOSES (Baroni et al., 2014), using previously-estimated predict vectors of 400 dimensions.[4] We also experimented with *word2vec* (Mikolov et al., 2013) vectors pre-trained (both with cbow and skip-gram) and both word2vec and GloVe with vectors trained on Qatar Living data, but we discarded them, as they did not contribute positively to our approach. Both q and c are then represented by a sum of the vectors corresponding to the words within them (neglecting the subject of c), and compute the cosine similarity to estimate $sim(q, c)$.

2.2 Context Simone/Alberto

Intuitively, whether a question includes further comments by u_q (some of them acknowledging), more than one comment from the same user, or whether

[3]Available at `http://nlp.stanford.edu/projects/glove/`; last visit: Jan 6th, 2015.

[4]Available at `http://clic.cimec.unitn.it/composes/semantic-vectors.html`; last visit: Jan 6th, 2015.

q belongs to a category in which a given kind of answer is expected, are important factors. Therefore, we consider set of features that try to describe a comment in its context.

Let $C = c_1, \ldots, c_C$ be the stream of comments associated to question q, asked by u_q. The first subset of features for comment c are of type boolean. Four of them are set to `True` according to the following criteria:

1. c is written by u_q (i.e. the same user behind q);
2. c is written by u_q and contains and acknowledgment (e.g. *thank**, *appreciat**);
3. c is written by u_q and includes further questions; and
4. c is written by u_q and includes no acknowledgments nor further questions.

The second subset intends to model c according to those comments by u_q appearing in its proximity. Intuitively, whether c appears close to an acknowledgment or further questions by u_q could be a relevant factor when classifying it. Our function to represent the relationship between a comment c_{t-k} in time $t-k$ and $c_{q,t}$, given that t is the time of the comment by u_q is as follows:

$$f(c_{t-k}) = \max\left(0 \ , \ 1.1 - (k * 0.1)\right) \tag{4}$$

where k is the distance between c_{t-k} and $c_{q,t}$ in the past. This function, which stop criterion is the occurrence of another comment by u_q, is applied to generate four features according to four criteria:

5. a c_q for which feature 2 is `True`,
6. a c_q for which feature 2 is `False`,
7. a c_q for which feature 3 is `True`, and
8. same as the previous one, but looking at the future instead.

We also tried to model potential dialogues by identifying interlacing comments between two users. Our dialogue features rely on identifying a sequence of comments

$$c_i \rightarrow c_j \rightarrow c_i \rightarrow c_j^*,$$

where u_i and u_j are the authors of c_i and c_j. Note that comments by other users can appear in between this “pseudo-conversation”. Three features are considered, whether a comment is at the beginning, middle, or ending position of the pseudo-dialogue. We consider three more features for those cases in which $q = j$.

We are also interested in realizing whether a user u_i has been particularly active in a question. As a result, we consider one boolean feature, whether u_i wrote more than one comment in the current stream, and three more features identifying the first, middle and last comments by u_i. One extra real feature counts the total number of comments written by u_i.

Qatar Living includes twenty-six different categories in which a person could request for information and advice. Some of them tend to include more open questions and even invite to discussions on ambiguous topics (e.g., *life in Qatar*, *Qatari culture*). Some others require more precise answers and allow for less discussion (e.g. *Electronics*, *visas and permits*). Therefore, we include one boolean feature per category to consider this information.

We empirically observed that the likelihood for a comment to be `GOOD` decreases the farther it appears from the question. Therefore, we consider one more real-valued feature: $\max(20, i)/20$, where i represents the position of the comment in the stream.

2.3 Word n-Grams

Our intuition is that a properly produced question should allow for the creation of `GOOD` comments. That is, objective and clear questions would tend to produce objective and `GOOD` comments. On the other side, subjective or badly formulated questions would call for `BAD` comments or even discussion (i.e. dialogues) among the users. When talking about comments, they could also include specific indicators that trigger a `GOOD` or `BAD` class, regardless of the specific question it intends to reply to.

Our aim is capturing those words or pairs of words which are associated to questions and comments in the different classes. Our features are composed of $[1, 2]$-grams by analyzing independently the question and comments. The weights are based on tf-idf on the whole Qatar Living dataset.

2.4 Heuristics

Exploring the data, we noticed that many GOOD comments suggested visiting a Web site or writing to 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, used in task B only, intend to determine whether a comment is positive or negative, which could be associated to YES or NO answers. A quantitative polarity of c is modeled as:

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

where $pol(w)$ represents the polarity of word w in the NRC Hashtag Sentiment Lexicon v0.1 (Mohammad et al., 2013).[5] Words with polarity in the range $(-1, 1)$ are discarded to neglect nearly neutral words.

We consider other boolean features on the existence of some keywords in the comment. Features are set to true if c contains (*i*) *yes*, *can*, *sure*, *wish*, *would* or (*ii*) *no*, *not*, *neither*.

2.6 User's Profile

With this set of features we aim at modeling 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 the user has produced before. We also consider the average word length of GOOD , BAD , POTENTIAL, and DIALOGUE comments. These features are computed both considering all the questions and only those from the same category as the current one.[6] Even if these features seem to fit with task A, rather than B, at development time they showed to be effective only for the latter one. Therefore, we only applied the user profiles to task B.

[5] http://www.umiacs.umd.edu/~saif/WebPages/Abstracts/NRC-SentimentAnalysis.htm; last visit: Jan 18, 2015.

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

ar	**DIRECT**	**RELATED**	**IRREL**	**MACRO**
prim	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**	**MACRO**
prim	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**	**MACRO**
prim	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 overall F_1-measure of our *prim*ary and *cont*rastive submissions to SemEval Task 3 for Arabic (A) and English (en). the

3 Submissions and Results

Following we describe our three primary submissions to the three subtasks. The contrastive submissions (two per task) are discussed later on. Table 1 includes the results obtained with these submissions.

3.1 Primary Submissions

In general, our approaches perform multiclass classification on the basis of a one-vs-rest support vector machines strategy (i.e. we train one classifier for each class). Our classifications for both Arabic and English A are made at comment level.

Arabic Our submission is based on a logistic regression approach. The utilized features are lexical similarities (Section 2.1) and n-Grams (Section 2.3), together with the predictions obtained with our contrastive submission 1 (cf. Section 3.2).

English A The submission applies the linear-kernel SVM for model estimation from scikit-learn.[7] We used a one-versus-all approach to account for the fact that the learning problem is a multiclass one. We tuned the C hyper-parameter of the SVM in order to deal with the class imbalance —by increasing the value of C, we built more complex classifiers for those classes with less instances.

The features for this submission consist of lexical and semantic similarities (Section 2.1), context information (Section 2.2), n-Grams (Section 2.3), and

[7] http://scikit-learn.org/stable/

heuristics (Section 2.4). In a sort of stacking, the output of our rule-based system from the contrastive submission 2 is included as another feature.

English B Following the strategy applied during the manual labelling of the YES/NO questions by the task organizers (Màrquez et al., 2015), our approach to task B is divided in three steps: (*i*) identifying the GOOD comments among those associated to the question; (*ii*) classifying each of the GOOD comments as YES, NO, or UNSURE; and (*iii*) aggregation. The overall answer to a question is that of the majority of the comments. In case of draw, we opt for labeling it as UNSURE.[8] Step (*i*) is indeed task A. As for step (*ii*), our approach to this task is identical as that for English A, but adding the features described in Sections 2.5 and 2.6.

Differently to the rest of submissions, our submitted results were obtained with a classifier trained on the training data only (the development set was neglected). The reason behind this decision was obtaining an unexpected distribution of (mostly) YES answers on the test set (completely different to that observed in both training and development partitions. Further experiments carried out after the submission demonstrated that the causes for such an unexpected behavior were a few features for which no statistics enough existed to be valuable as well as a buggy implementation of some of the other features. Further discussion is included in Section 4.3.

3.2 Contrastive Submissions

Arabic Our contrastive submission 1 consists of a ranking problem. The similarity $sim(q, c)$ for every c in the question is computed, after stopwording and stemming, as

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

where $\omega(t)$ is the empirically-set weight of an n-gram: $\omega = 1$ for 1-grams and $\omega = 4$ for 2-grams. Given the 5 comments $c_1, \ldots, c5 \in C$ associated to q, the maximum similarity $\max_C sim(q, c)$ is mapped to a maximum 100% similarity and the rest of the scores are mapped 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.

As for the contrastive submission 2, we built a binary classifier based on logistic regression: DIRECT or no. The comments are then sorted according to the classifier's prediction confidence and the final labels are assigned accordingly: DIRECT for the 1st ranked, RELATED for the 2nd ranked, and IRRELEVANT for the rest. Only lexical similarities are included as features (discarding those weighted with idf variants).

English A For our contrastive submission 1, the same machine learning schema as for the primary submission is used. In contrast to the primary submission, this time the C hyper-parameter is set to the default value and the class imbalance was dealt by tuning the j parameter (cost of making mistakes on positive examples). In this case we use SVM$^{\text{light}}$ (Joachims, 1999).

Our English contrastive submission 2 operates in the same way as the Arabic contrastive submission 1. The applied ranges are the same, this time used to assign the classes GOOD, POTENTIAL, and BAD. Some heuristics override the so generated decisions: 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*). Comments written by the author of the question or including acknowledgments are considered DIALOGUE; which become BAD comments.

English B Our contrastive submission 1 is identical as the primary one, but considering both training and development data for estimating the model.

The contrastive submission 2 consists of a rule-based system. A comment is labeled as YES if it starts with affirmative words: *yes*, *yep*, *yeah*, etc. It is labeled as NO if it starts with *no*, *nop*, *nope*, etc,

3.3 Discussion and Further Experiments

English B

- Any proposal? Probably no questions enough for an error analysis

Arabic

[8] YES, the majority class in the training and development partitions, could have been the default answer. Still, we opted for a conservative decision: deciding UNSURE if no evidence enough was at hand.

Subm. without	DIR	REL	IREL	MACRO
n-grams				76.75
cont_1				67.74
no max iyas				79.13
no max sim				78.37
no max iyas sim				78.69

Table 2: Post-competition experiments Arabic

Only with	GOOD	BAD	POT	MACRO
context				47.90
n-grams				44.86
heuristics				52.57
Similarities				46.16
lexical				44.82
syntactic				36.47
semantic				42.16
Without	**GOOD**	**BAD**	**POT**	**MACRO**
context				51.49
n-grams				55.17
heuristics				48.60
Similarities				??.??
lexical				53.34
syntactic				53.73
semantic				53.50

Table 3: Post-competition experiments English A

- Let's ask the Arabic team!

4 Pots-Submission Experiments

4.1 Arabic

Table 2 shows the results obained after discarding each of the different features families.

4.2 English Task A

Table 3 shows the results obtained on the test set by considering both the different subsets of features in isolation (*only with*) or all the features except for a subset (*without*). According to these figures, the heuristic features seem to be the most useful, followed by the context-based information. The best performing subsets are close to that combining all the features (cf. Table tab:results).

4.3 English Task B

After the submission we investigated on the reasons why learning on the training only was considerably better than learning on the union of the training and development sets. The sequences of predicted target labels on the test set in the two learning scenarios showed considerable differences: when learning on the union of the training and development sets the predicted labels were YES on all but three cases. After correcting a bug, the results obtained by learning on the union of the training and development sets were the ones in the "after_1" submission in the first row of Table 4, i.e. a Macro F1 value of 51.98. Learning on the training set only still gives a higher macro F1 of 69.35, but the sequences of predicted labels are now more consistent and the difference might not be significant (REFERENCE TO TASK DESCRIPTION PAPER). We observed that the values of those features counting the number of Good, Bad, and Potential comments within categories from the same user (cf. Section **??**) vary greatly when computed on the training or training+dev datasets. This is due to the fact that the number of comments of a user for a category is, in most cases, too limited to generate reliable statistics. After discarding these three features, the obtained Macro F1 value is 55.95 (see "after_2" submission in Table 4), which represents a higher performance than the ones obtained during at submission time.

Subm.	YES	NO	UNSURE	MACRO
after_1	78.79	57.14	20.00	51.98
after_2	85.71	57.14	25.00	55.95

Table 4:

References

L Allison and T I 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)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

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, pages 7–16, Uppsala, Sweden, July. Association for Computational Linguistics.

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.

Caroline Lyon, James Malcolm, and Bob Dickerson, 2001. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, chapter Detecting Short Passages of Similar Text in Large Document Collections.

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 2015)*. Association for Computational Linguistics.

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*, pages 746–751, Atlanta, GA, June. Association for Computational Linguistics.

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 seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, GA, June.

Alessandro Moschitti. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In Frnkranz, Johannes and Scheffer, Tobias and Spiliopoulou, Myra, editor, *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*, pages 64–71.

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)*, pages 1532–1543. Association for Computational Linguistics.

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

Michael J. 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. ACM.