bunji at SemEval-2017 Task 3: Combination of Neural Similarity Features and Comment Plausibility Features

Yuta Koreeda[1], Takuya Hashito[2], Yoshiki Niwa[1], Misa Sato[1],
Toshihiko Yanase[1], Kenzo Kurotsuchi[1], and Kohsuke Yanai[1]

[1]Research & Development Group, Hitachi, Ltd.
[2]Industry & Distribution Business Unit, Hitachi, Ltd.
{yuta.koreeda.pb, takuya.hashito.qk, yoshiki.niwa.tx,
misa.sato.mw toshihiko.yanase.gm,
kenzo.kurotsuchi.qs, kohsuke.yanai.cs}@hitachi.com

Abstract

This paper describes a text-ranking system developed by bunji team in SemEval-2017 Task 3: Community Question Answering, Subtask A and C. The goal of the task is to re-rank the comments in a question-and-answer forum such that useful comments for answering the question are ranked high. We proposed a method that combines neural similarity features and hand-crafted comment plausibility features, and we modeled inter-comments relationship using conditional random field. Our approach obtained the fifth place in the Subtask A and the second place in the Subtask C.

1 Introduction

This paper explains the participation of the bunji team in SemEval-2017 Task 3 on Community Question Answering (CQA) (Nakov et al., 2017), Subtask A and Subtask C. The goal of the task is to re-rank the comments in a question-and-answer forum such that useful comments for answering the question are ranked high. Subtask A is extraction of relevant answers from comments in a question thread. Given a question and its comments, the system must re-rank the comments according to their relevance with respect to the question. Subtask C is extraction of relevant answers from comments in different question threads. Given a question (the original question), questions that are possibly related to the original question (the relevant questions) and comments to the relevant questions, the system must re-rank the comments according to their relevance with respect to the original question. Since the task is ranking, the primary metric is mean average precision (MAP).

Our model consists of three elements; use of similarity features, use of comment plausibility features and a supervised scoring method that models inter-comments relationship. The similarity features are designed to capture the similarities between a question and a comment because a valid answer should be on the same topic as the question. Similarity features were utilized by many teams in SemEval-2016 (Nakov et al., 2016). In this work, we take a deep learning approach to extract similarity features.

The comment plausibility features are designed to capture characteristics that relevant answers tend to have. Similar concept was proposed by Mihaylova et al. (2016), who tried to model readability, credibility, sentiment and trollness. The comment plausibility features were hand-crafted to incorporate human knowledge about CQA.

In past CQA tasks, some teams incorporated inter-comments relationship. An example of such relationship is acknowledgement, where a good answer is likely to be followed by acknowledgement of the questioner. Barrn-Cedeo et al. (2015) modeled inter-comments relationship by taking distance to nearest acknowledgement as a feature and using Conditional Random Field (CRF) to model transition probability between relevant and irrelevant comments. In our work, we try to model inter-comments relationship in much simpler way; by concatenating features of adjacent comments and by utilizing CRF for final ranking function.

2 Method

Our proposed method is constructed in following steps:

(i) Neural network is trained to extract similarity features independently to the rest of the system,

(ii) comment plausibility features are extracted with hand-crafted rules,

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 353–359,
Vancouver, Canada, August 3 - 4, 2017.

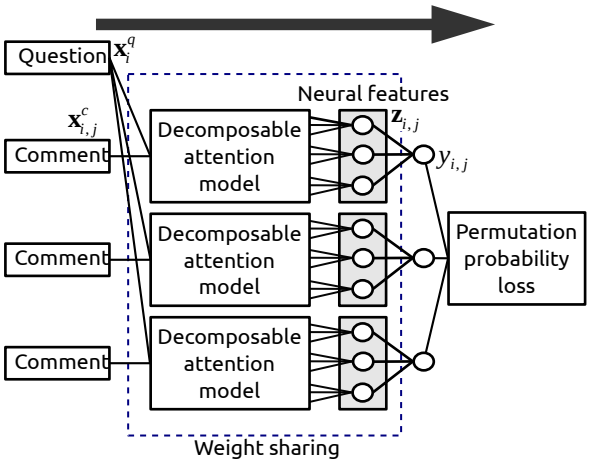

Figure 1: Neural feature extraction

(iii) neural similarity features and comment plausibility features are concatenated to form the combined features, and
(iv) CRF is optimized on the combined features while keeping the neural network fixed.

We used almost identical method for Subtask C. The differences in the system for Subtask A and for C are discussed in Section 2.4.

2.1 Neural Similarity Features

One of the challenges in the CQA task is that question and comment texts tend to be long. This makes use of recurrent neural network difficult, because recurrent neural network is known to be less effective against a long sequence (Lai et al., 2015). In this work, we make assumption that only a very small region of a question and a comment is needed to decide whether the comment is relevant. For example, given a 62-words question,

> *... and would like to know the typical business dress code in Doha for Non Nationals. Is it OK for men to wear short sleeve shirts? For women; I am assuming the more conservative; ...*[1]

and a 50-words comment,

> *I agree with MR M;its not much to worry of your dress.. its not an issue over here ;just be modest...*[1]

We only need underlined parts of the question and the answer to identify that the comment is relevant.

[1]From SemEval-2017 data (Nakov et al., 2017) (http://alt.qcri.org/semeval2017/task3/index.php?id=data-and-tools)

We propose a feature extraction method based on a decomposable attention model (Parikh et al., 2016). This method is designed to model alignment between two sequences of text, allowing the system to jointly identify informative region and predict whether the comment is relevant.

The overview of our neural network is shown in Figure 1. Each question-comments thread (one question and multiple comments) is mapped to a real value score using a decomposable attention model. The loss for stochastic gradient decent is calculated for each thread using list-wise ranking loss.

As preprocessing, we remove HTML tags, apply tokenization and lowercase all characters. Named entities, image tags, URLs and numerics are each converted to special symbols. A question subject text is prepended to the corresponding question text. We truncate question and comment text to first 50 tokens.

The c-th token of j-th comment text ($1 \leq j \leq N$) in i-th thread is then mapped to word vector representation $\mathbf{x}_{i,j,c}^C \in \mathbb{R}^M$ and the q-th token of the question text in i-th thread to $\mathbf{x}_{i,q}^Q \in \mathbb{R}^M$. The word vector was pretrained with the raw forum text provided by the organizers which contained approximately 100 million words.We only use 50,000 most frequent words and the rest of the words are mapped to an averaged vector of 50 least frequent words.

Each combination of a comment $\mathbf{x}_{i,j}^C = \{\mathbf{x}_{i,j,c}^C\}_{1 \leq c \leq L^C}$ and a question $\mathbf{x}_i^Q = \{\mathbf{x}_{i,q}^Q\}_{1 \leq q \leq L^Q}$ is mapped to a question-comment vector $\mathbf{z}_{i,j}$ using the decomposable attention model. First, the model compares and calculates attention $e_{i,j,c,q}$ for each token combination,

$$e_{i,j,c,q} = F\left(\mathbf{x}_{i,j,c}^C\right)^T F\left(\mathbf{x}_{i,q}^Q\right), \quad (1)$$

where F is a feed forward neural network. Then, the model extracts subphrase of $\mathbf{x}_{i,j}^C$ that is soft-aligned against $\mathbf{x}_i^Q$ using attention mechanism:

$$\bar{e}_{i,j,c,q}^C = \frac{\exp(e_{i,j,c,q})}{\sum_{s=1}^{L^C} \exp(e_{i,j,s,q})} \quad (2)$$

$$\chi_{i,j,q}^C = \sum_{c=1}^{L^C} \bar{e}_{i,j,c,q}^C \mathbf{x}_{i,j,c}^C, \quad (3)$$

Then we compare the word vector to the soft-aligned subphrase and aggregate all the combina-

tions:

$$\mathbf{v}^Q_{i,j} = \sum_{q=1}^{L^Q} G([\chi^C_{i,j,q}, \mathbf{x}^Q_{i,q}]), \quad (4)$$

where G is a feed forward neural network and $[\bullet, \bullet]$ denotes the concatenation of vectors. This is calculated vice-versa for $\mathbf{v}^C_{i,j}$. Finally, we map $\mathbf{v}^Q_{i,j}$ and $\mathbf{v}^C_{i,j}$ to a score $y_{i,j} \in \mathbb{R}$:

$$\mathbf{z}_{i,j} = H([\mathbf{v}^Q_{i,j}, \mathbf{v}^C_{i,j}]) \quad (5)$$

$$y_{i,j} = \sigma(\mathbf{z}_{i,j}) \cdot \mathbf{W} + \mathbf{b}, \quad (6)$$

where H is a feed forward neural network, σ is an activation function, and $\mathbf{W}$ and $\mathbf{b}$ are model parameters. The representation $\mathbf{z}_{i,j}$ is used as the neural features, which is combined with comments plausibility features to form our final model.

The scores $\mathbf{y}_i = \{y_{i,j}\}$ are optimized to predict ground truth label sequence $\mathbf{t}_i = \{t_{i,j}\}$ with permutation probability loss (Cao et al., 2007). A ground truth label is set 1 if it is labeled "Good" and 0 if it is labeled "PotentiallyUseful" or "Bad" in accordance to the task rules (Nakov et al., 2017). We use $k = 1$ permutation probability distribution function $P : \mathbb{R}^N \mapsto \mathbb{R}^N$, such that

$$P(\mathbf{y}_i) = \left[\frac{\exp(y_{i,j})}{\sum_n^N \exp(y_{i,n})} \right]_{j \in \{1,2,\ldots,N\}}. \quad (7)$$

The permutation probability loss is defined as $D_{KL}(P(\mathbf{t}_i) \| P(\mathbf{y}_i))$ where D_{KL} is Kullback-Leibler divergence between two distributions.

Since decomposable attention model and permutation probability loss are fully differentiable, we can optimize the whole network with mini-batch stochastic gradient descent with backpropagation. We use rmsprop with momentum (Graves, 2013) and learning rate reduction of 1% for every 100 batches. Dropout (Srivastava et al., 2014) and L2-norm regularization are applied to each layer of feed forward neural network to avoid overfitting. Batch normalization (Ioffe and Szegedy, 2015) is applied and gradient norm is clipped to 5.0 to improve the training stability. We use leaky rectified linear unit for activation function σ as shown in Equation (8) to stabilize the training.

$$\sigma(x) = \max(x, 0.2x) \quad (8)$$

Other hyperparameters are shown in Table 1. Above model selection and hyperparameters were manually tuned by validation against SemEval-2016 test data.

Parameter	Subtask A	Subtask C
Number of layers in F and H	3	3
Number of layers in G	3	3
Dimension of $z_{i,j}$	200	50
Dimensions of other layers	200	200
Word vector dimension M	200	200
Dropout rate	0.1	0.1
L2 regularization coefficient	0.1×10^{-4}	0.2×10^{-4}
L2 regularization coefficient for W	0.2×10^{-4}	0.3×10^{-5}
Initial learning rate	0.5×10^{-5}	0.5×10^{-5}
Mini-batch size	2	1
Max tokens	50	30
Training epochs	50	30

Table 1: Hyperparameters of the neural network

?, !, what, which, who, where, when, why, whom, how, hi, ⟨what, which, who, where, when, why, whom, how, hi⟩, ⟨yes, yep, year⟩, ⟨no, nope, nah⟩, ⟨thank, thanks, tnx, thx⟩, ⟨you, u⟩, ⟨good, greate, nice⟩, ⟨bad, not, non⟩

Table 2: Lexicons used in function-of-a-comment features. $\langle\bullet\rangle$ denotes a feature that is positive when any of the words are present in the comment.

2.2 Comment Plausibility Features

Comment plausibility features are designed to extract information that is not captured by neural similarity features. These features are divided into five groups: (1) function of a comment, (2) answer adequacy, (3) dialog structure, (4) answerer's meta-information, and (5) miscellaneous.

Part of speech tagging and named entity recognition for comment plausibility features are carried out using Stanford CoreNLP (Manning et al., 2014).

2.2.1 Function of a Comment

This group of 39 features is designed to capture the function of a comment; e.g. trying to answer the question, making remarks, or asking the questioner for more information. This group of features is extracted from each comment.

The occurrence of each word in Table 2 within each comment is extracted as a binary feature. We use the part of speech tag for the first and the final word of the comment. This is expressed as one-hot representation of whether the first/final word is noun, adjective, adverb, verb, auxiliary verb, conjunction (for the final word only) or interjection. We also added a feature whether the first word is "is."

We also use ratio of each part of speech tag to the number of tokens.

#	Comparing			IDF source					
				Per thread			All dataset		
	S	Q	C	S	Q	C	S	Q	C
1		○	○		○	○			
2		○	○	○	○	○			
3	○		○	○		○			
4	○		○	○	○	○			
5		○	○					○	○
6		○	○				○	○	○
7	○		○				○		○
8	○		○				○	○	○

S = question subject, Q = question text, C = comment text

Table 3: Types of TF-IDFs for calculating cosine distance. Column *Comparing* shows text blocks to extract and compare TF-IDF. Column *IDF source* shows the documents used to calculate IDF, where each text block is regarded as a single document.

2.2.2 Answer Adequacy

This group of 27 features is designed to capture whether the comment has adequate information to answer the question. For this purpose, this group of features is extracted from each question-answer pair.

The presence of each word (what, which, who, where, when, why, whom, how, hi, and any of do, does, or did) within a question is extracted as a binary feature. The presence of each type of named entities (location, person, organization, money, percent, date and time) and the presence of any type of the named entities, numerics, image tags and URLs in each comment are also extracted.

The relative length of a comment to a question is also extracted. This is based on the idea that the answer tends to be long when a question is long. This relative length is calculated for 6 variants; i.e. the number of words/characters in a comment divided by,

(i) the total number of words/characters in the question and the comment,

(ii) the total number of words/characters in the question subject, text and the comment text, and

(iii) the total number of words/characters in the question subject and the comment text.

2.2.3 Dialog Structure

This group of four features is designed to capture the dialog structure of comments. For this purpose, this group of features is extracted for each comment using the whole thread.

Dialog structure features include the binary features for each of the following statements:

(i) If the comment is posted by the question author.

(ii) If the comment contains the name of the question author.

(iii) If the comment contains a name of the user other than the question author (comment contains a string with "@" prefix).

We use reciprocal chronological order (e.g. $1/3$ for the third comment) to capture the global position of a comment.

2.2.4 Answerer's Meta-information

This group of two features is designed to capture the answerer's meta-information. For this purpose, this group of features is extracted for each comment using the whole dataset.

For example, whether a comment is written by the author of the question is important information because he or she hardly ever knows the answer.

Answerer's meta-information features are binary features for each of the following statements:

(i) If the comment author is anonymous.

(ii) If the comment author has posted a comment elsewhere in the dataset.

2.2.5 Miscellaneous

To further improve the performance, we adopted a lexicon of 23 words with the lowest semantic orientation in CQA (Balchev et al., 2016) and extracted the occurrence of these words from the comments.

We also use the cosine distance between the term frequency-inverse document frequency (TF-IDF) vectors of a question and a comment. We use eight types of TF-IDF as listed in Table 3, each characterized by document blocks (question subject, question text or comment text) to compare and to calculate IDF. We also used presence of word overwrap in the question-comment and the subject-comment pair as binary features. While redundant to neural similarity features, redundant features increase the overall performance by acting like an ensemble.

We use the cosine distance between the TF-IDF vector of a comment and an averaged TF-IDF vector of all comments in thread. This is extracted for an averaged TF-IDF vector of all comments in dataset, as well. These features are intended to capture amount of distinctive information that each comment contains.

Submission	Subtask A				Subtask C			
	Position	MAP	AvgRec	MRR	Position	MAP	AvgRec	MRR
Primary*	5	86.58	92.71	91.37	2	14.71	29.47	16.48
Contrastive 1†	7	85.29	91.77	91.48	5	8.19	15.12	9.25
Contrastive 2‡	7	84.01	90.45	89.17	1	16.57	30.98	17.04
Top team	1	88.43	93.79	92.82	1	15.46	33.42	18.14
baseline(IR)	-	72.61	79.32	82.37	-	9.18	21.72	10.11

* Combined † Comment plausibility features ‡ Neural features

Table 4: 2017 official result

Submission	Subtask A		Subtask C	
	2017	2016	2017	2016
Primary	86.58	75.6	14.71	39.9
Contrastive 1	85.29	74.4	8.19	38.0
Contrastive 2	84.01	71.4	16.57	28.0
Top team	88.43	79.2	15.46	55.4
IR baseline	72.61	59.5	9.18	40.3

Table 5: Comparison of MAP scores in 2016 and 2017 test dataset

2.3 Combined features

The neural features and the comment plausibility features are concatenated to form Primary run for Subtask A and C. The features are further extended by concatenating features from two comments before and after the target comment, resulting in concatenated features over five comments. This allows extending the dialog structure features (Section 2.2.3) without adding too many features, as described in Section 1.

We use first order linear CRF by regarding each comment as an observation and a thread as a sequence. Along with concatenated features, CRF allows non-local optimization of inter-comments relationship. For example, presence of "yes" after a good answer is likely to be acknowledgement by the questioner. In this case, effect of "yes" is conditioned on the label of the previous comment.

CRF is trained using L-BFGS with L1 regularization coefficient of 1.0 and L2 regularization coefficient of 0.001. We use CRFsuite (Okazaki, 2007) as an implementation of CRF.

2.4 Modification for Subtask C

For neural similarity features, hyperparameters were manually tuned for Subtask C as shown in Table 1. On training neural models for Subtask C, we added all the question-comment pairs from Subtask A to augment the data.

For comment plausibility feature, we applied greedy stepwise backward elimination using SemEval-2016 test data as validation data. We tested the deletion of each feature and removed the feature whose deletion gives the best MAP improvement. We repeated the process until MAP no longer improves. The process removed following features:

(i) Presence of any of words ⟨what, which, who, where, when, why, whom, how, hi⟩.
(ii) The relative length of a comment (Section 2.2.2, (ii)).
(iii) Reference to the question author (Section 2.2.3, (i) and (ii)).
(iv) Answerer's meta-information.
(v) TF-IDF (Table 3, #1 and #4).

3 Experiments

Our Primary submission was CRF with combined features. Contrastive 1 was CRF with only the comment plausibility features. Contrastive 2 was CRF with only the neural similarity features.

The official results for the 2017 test data are shown in Table 4. The Primary obtained the fifth and the second in Subtask A and C, respectively.

The combined features (Primary) was much better than Contrastive 1 and 2 in Subtask A, as expected. The large increase of 1.29 MAP score from Contrastive 1 to the Primary implies that the neural features and comment plausibility features were capturing different aspects of the problem.

On the other hand, Contrastive 1 performed poorly in Subtask C. This was partially because the similarity was more important in Subtask C, which contained many unrelated comments. Thus neural similarity features performed much better than in Subtask A and comment plausibility feature did much worse. Another reason for Contrastive 1's poor performance may have been due to the over-fitting to development dataset, as implied by large performance drop from 2016 dataset (Table 5).

Feature	MAP
All features	76.23
– Author's comment or not	74.80
– Reciprocal of answer's number	75.00
– Word "?"	75.64
– First word is an auxiliary verb	75.65
– Word "avatar"	75.66
– Word "whom"	75.70
– First word is adjective	75.72
– TF-IDF (Table 3, #1)	75.73

Table 6: The top 8 contributing comment plausibility features in Subtask A

Feature	MAP
All features	38.70
– Word "do," "does" or "did"	37.24
– Word "who"	37.63
– Word "fs"	37.93
– Final word is an adverb	38.13
– Word "what" in the comment	38.23
– First word is a noun	38.33
– Word "?"	38.35
– Cosine distance between a comment TF-IDF and an averaged TF-IDF over all comments in thread	38.39

Table 7: The top 8 contributing comment plausibility features in Subtask C

To identify the contributing features within the comment plausibility features, we carried out additional experiments on 2016 test dataset where we eliminated each feature one by one from the Primary system. The top 8 contributing features are shown in Table 6 (Subtask A) and 7 (Subtask C). From the result, the comment plausibility features seem to work as a blacklist for comments that are unlikely to be an answer. For example, occurrence of words "?," "do," "does," "did," and "what" all contribute to identifying a question which are less likely to be a comment.

Our neural similarity feature performed worse than the previous application of recurrent neural network to Subtask A (MAP scores of 75.7 against our 71.4) and to Subtask C (MAP scores of 47.2 against our 28.0) (Wu and Lan, 2016). The reason for the inferior performance may be due to very large vocabulary of CQA, which caused the neural network to fall back to only using commonly appearing words in many cases. As a supporting observation, attention weight seem to localize on very few commonly appearing words instead of on more meaningful region of text (Figure 2). Use of sub-word vocabulary can help overcome this problem (Yoon Kim et al., 2016; Wu et al., 2016). Also, we manually tuned the hyperparamters for neural network. Random searching for better hyperparameters can improve the overall performance.

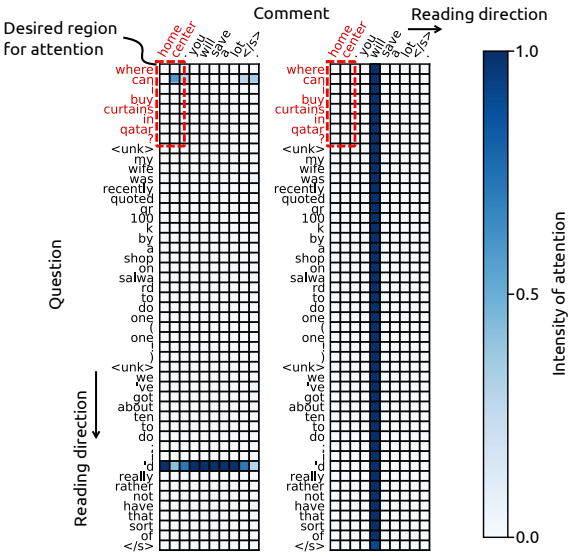

Figure 2: Visualization of attention ($\bar{e}^{Q}_{i,j,c,q}$ on left and $\bar{e}^{C}_{i,j,c,q}$ on right) in failing case. Attention had concentrated on commonly appearing words rather than more informative regions.

4 Conclusions

This paper explains the participation in SemEval-2017 Task 3, Subtask A and Subtask C, which is a problem of ranking the comments in community question answering forum according to their relevance to the question. We proposed a method that combines neural similarity features and comment plausibility features, and modeled inter-comments relationship. Our approach obtained the fifth place in the Subtask A and the second place in the Subtask C.

For future work, we will improve the neural method so that it can better handle large vocabulary of CQA. We will also incorporate systematic end-to-end tuning on both feature selection and neural method to deal with over-fitting problem.

References

Daniel Balchev, Yasen Kiprov, Ivan Koychev, and Preslav Nakov. 2016. PMI-cool at SemEval-2016 Task 3: Experiments with PMI and Goodness Polarity Lexicons for Community Question Answer-

ing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pages 844–850.

Alberto Barrn-Cedeo, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Llus Mrquez, Preslav Nakov, and Alessandro Moschitti. 2015. Thread-Level Information for Comment Classification in Community Question Answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. pages 687–693.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning*. pages 129–136.

Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *arXiv* .

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*. volume 37, pages 448–456.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. pages 2267–2273.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60.

Tsvetomila Mihaylova, Pepa Gencheva, Martin Boyanov, Ivana Yovcheva, Todor Mihaylov, Momchil Hardalov, Yasen Kiprov, Daniel Balchev, Ivan Koychev, Preslav Nakov, Ivelina Nikolova, and Galia Angelova. 2016. SUper Team at SemEval-2016 Task 3: Building a Feature-Rich System for Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pages 836–843.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pages 27–48.

Preslav Nakov, Llus Mrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pages 525–545.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (CRFs). http://www.chokkan.org/software/crfsuite/.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 2249–2255.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Guoshun Wu and Man Lan. 2016. ECNU at SemEval-2016 Task 3: Exploring Traditional Method and Deep Learning Method for Question Retrieval and Answer Ranking in Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pages 872–878.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Technical report.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. Character-Aware Neural Language Models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. pages 2741–2749.