A Hybrid Approach to Question-answering for a Banking Chatbot on Turkish: Extending Keywords with Embedding Vectors

Enes Burak Dündar, Tolga Çekiç, Onur Deniz and Seçil Arslan
R&D and Special Projects Department, Yapi Kredi Technology, Istanbul, Turkey

Keywords: Question-answering, Chat Bot, Word Embeddings.

Abstract: In this paper, we have proposed a hybrid keyword-word embedding based question answering method for a Turkish chatbot in banking domain. This hyrid model is supported with the keywords that we have determined by clustering customer messages of Facebook Messenger bot of the bank. Word embedding models are utilized in order to represent words in a better way such that similarity between words can be meaningful. Keyword based similarity calculation between two questions enhanced our chatbot system. In order to evaluate performance of the proposed system, we have created a test data which contains questions with their corresponding answers. Questions in the test are paraphrased versions of ones in our dataset.

1 INTRODUCTION

Automated Q&A or troubleshooting systems have been gaining increasing popularity in recent years. These systems can be used as personal assistants or domain based Q&A systems. This work proposes a keyword-based Q&A system to complement our previous work in creating a banking domain chatbot for Turkish language (Soğancıoğlu et al., 2017). General architecture of our chatbot is given in Figure 1. Each day, thousands of customers contact the bank to convey their demands, complaints or ask general questions about products or services. Customers utilize different channels such as customer support phone or chat, social media, etc. and a high amount of human resource is required to answer all these questions. Thus, developing a Q&A system that can reply a wide range of customer queries is vital for a bank to have an efficient customer support.

In order to create a banking Q&A system that can help customers with their most common problems or requests reliably, firstly most frequently asked questions are identified using two unlabeled datasets. Then after answers for these most common questions are determined by experts, a dataset consisting of question and answer pairs is created. Then a retrieval based question answer system is designed. The system works by matching user queries with questions in the dataset. The answer of the most similar question is returned.

Initial experiments with sentence similarity measures such as Word Mover's Distance (WMD), and Q-gram similarity fell short of desired success rates. In order to address this issue, question-answer pairs are reviewed and it is seen that some words appear frequently. Thus, a keyword based similarity scheme is designed. Keywords are extracted from dataset via clustering. However, since Turkish is an agglutinative language every word can have arbitrary amount of various suffixes that can influence the meaning of the words. In order to handle various inflections of our keywords with different suffixes, word embeddings are used to discern keywords with suffixes.

Experiments have been performed on an annotated dataset of 138 questions. These questions are paraphrases and alternative forms of question answer pairs dataset and they have specific answers from the dataset. Our first set of experiments focused on accurately finding correct answers for questions. Secondly, experiments were made to gauge the success of system's understanding of not being able to find the correct answer since in a Q&A system it would be preferable to ask for a clarification rather than returning an unrelated answer. In this second set of experiments answers that admit not understanding the question are counted as positives while answers unrelated to question are counted as negatives. In experiments, our keyword based method have performed better than other sentence similarity methods.

Our contribution in this paper is the keyword based question answer system that can deal with suffixes intrinsically in mophologically rich languages such as

Dündar, E., Çekiç, T., Deniz, O. and Arslan, S.
A Hybrid Approach to Question-answering for a Banking Chatbot on Turkish: Extending Keywords with Embedding Vectors.
DOI: 10.5220/0006925701710177
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 1: KDIR*, pages 171-177
ISBN: 978-989-758-330-8

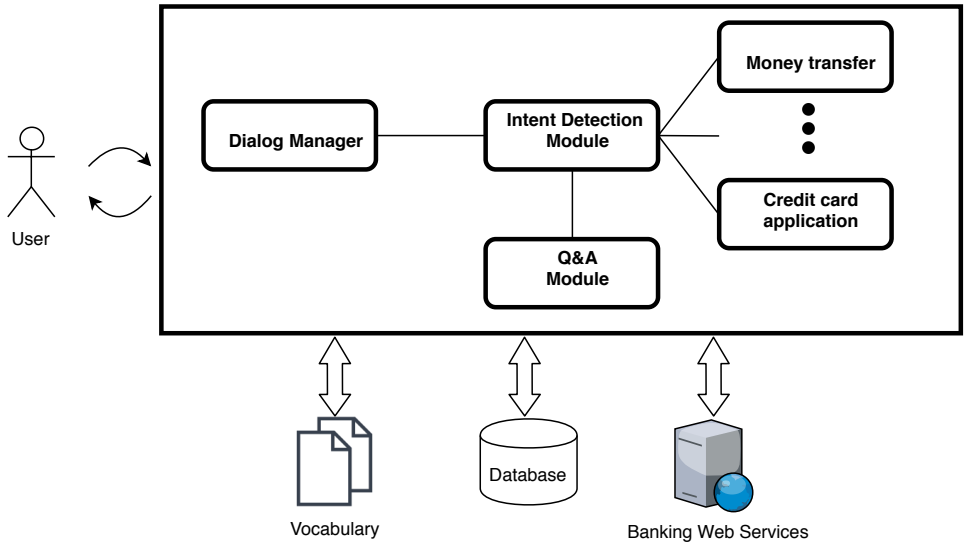

Figure 1: Chatbot Architecture.

Turkish. The rest of the paper is organized as follows. In Section 2, related works have been introduced. In Section 3, we have proposed our question-answering system on the banking domain. In Section 4, experimental results have been denoted. In Section 5, we summarize the proposed method, and denote possible future works.

2 RELATED WORKS

This study is performed to enhance the Q&A aspect of chatbot in our previous work in (Soğancıoğlu et al., 2017). This banking chatbot can perform tasks that require back-and-forth conversations between customer and the agent, such as taking a credit card application or money transfer while understanding the intent with the methodology explained in (Soğancıoğlu et al., 2016) and shown in 1. Chatbot also targets to offer Q&A troubleshooting for customer questions that doesn't require in depth conversations, which is treated as a retrieval task in this study. For Q&A systems, the state of the art model developed for SQuAD dataset is (Yu et al., 2018). The SQuAD is the Stanford question answering dataset which is a benchmark in which a lot of studies are ranked.

Turkish Q&A Systems. There have been previous work in Turkish Q&A systems. In one study, a question answer system is designed to be used in online courses for students to easily get their queries answered from course material (Yurekli et al.,). This system works like a search engine treating course material as pages and returns the most relevant section to the user. Another Turkish question answer system proposed uses name entity recognition and pattern matching to choose an appropriate reply (Çelebi et al., 2011). Patterns are used to categorize questions into one of nine question types and then with the help of name entity recognition the most likely answer is retrieved.

Word Embeddings. Word embeddings are representations of words in a high dimensional space. While one of the most popular models is word2vec (Mikolov et al., 2013) which showed that words can be represented in high dimension by keeping semantic and syntactic relationships, (Rumelhart et al., 1986) and (Bengio et al., 2003) are considered as earlier studies. In Section 3.3, a detailed information of word embeddings is given.

Sentence Similarity Measures. Q-gram similarity is the first text similarity measure reviewed for this work (Ukkonen, 1992). Q-gram is a similarity metric based on character-level string operations. At first, $n-1$ special characters are inserted at the beginning and end of two words which we want to evaluate their similarity. Then, they are divided into substrings with length n. Then, the similarity between two strings is denoted in Equation 1 in which A and B are set of substrings. Experiments in a previous work also showed that this measure is good for morphologically rich languages such as Turkish and noisy texts because of its character-level operations (Soğancıoğlu et al., 2017).

$$Q-gram-similarity = \frac{2*|A \cap B|}{|A \cup B|} \quad (1)$$

Another text similarity measure this work is based on is Word Mover's Distance (WMD) (Kusner et al., 2015). WMD makes use of word embeddings. In order to calculate a distance between two texts, each

word is compared with each word from the other text. Then the distance between most similar ones are added to get the total distance.

3 METHODOLOGY

3.1 Datasets

Two datasets obtained from two channels has been used in this study. First is from Webchat, which is a system our company offers customers to solve their problems via contacting a human agent. Hence, Webchat dataset is the collection of dialogues with messages written between customers and human agents. The webchat dataset includes dialogues from 2014 to 2016. This dataset contains more than a million dialogues and total number of sentences in all dialogues is over 13 million.

The second dataset is obtained from Facebook Messenger. This dataset is similar to Webchat, but here customers interact with a primitive troubleshooting chatbot that can reply to basic customer queries. This basic chatbot is previously developed as a quickwin solution - a decision tree based on keywords. Facebook messenger dataset only contains messages written by customers. Moreover, it is observed that average length of messages is shorter compared to messages in Webchat dataset. One possible reason might be due to the fact that customers, being aware that responses are given by an automatized system instead of a real human, shorten their queries. Therefore, messages in Facebook messenger dataset represent the summary of problem that customers want to ask.

Both datasets includes free written text messages from users and they contain high amount of typos, misspellings and grammatically wrong sentences. Therefore, since datasets consist of noisy texts, preprocessing and using word vectors for calculating similarity between words is very important.

3.2 Preprocessing

Messages in both datasets introduced in Section 3.1, contain Turkish and English letters. Therefore, Turkish letters are converted to their closest English counterparts for consistency since most of the letters belong to English. *ğ, ç, ş, ü, ö,* and *ı* are converted to *g, c, s, u, o,* and *i* respectively. What is more, spell checker is also utilized to find correct forms of words. For this approach, Zemberek (Akın and Akın, 2007), a natural language processing tool for Turkish language, has been utilized.

Before calculating similarity between two questions, they are preprocessed in following ways. At first, all punctuation marks are removed. Then, each letter is converted to lowercase. Finally, Turkish characters are converted to corresponding English words. Latter process is mainly done to ensure consistency because inspection of datasets show people can write with either a Turkish keyboard or an English keyboard and often Turkish characters are replaced with their English counterparts.

3.3 Word Embeddings

In order to calculate semantic similarities among words word embeddings are used. Word embeddings are dense vector representations of words. Word embeddings are used to get semantic similarities between synonymous words or same words with different suffixes. While a bag-of-words approach would treat inflectional forms and synonyms of words as entirely irrelevant, a well trained word embedding should capture similarity between such words. Furthermore, word embeddings can calculate vectors for frequent misspelling of words that are very similar to vector for original words. Such the closest vector for the word 'kredi' (credit in English) is 'kreid' which is a common typo of that word in datasets. We used two algorithms to train word vectors mainly due to their proven success in capturing semantic relations between words: Word2Vec and Fasttext.

Word embeddings with both methods have been trained on webchat dataset with all messages concatenated as a whole document. Since webchat dataset is domain specific, training word vectors on it provided meaningful semantic representations of banking related terms. Dimensions of the trained vectors are set as 300 since it is shown that 300 dimensional word vectors performs better in (Sen and Erdogan, 2014).

3.3.1 Word2vec

Word2Vec is a method for training word embeddings using neural networks (Mikolov et al., 2013). Word2Vec actually has two different ways to train embeddings: skip-gram and continous-bag-of-words(CBOW). Both versions of the method has a sliding window through words. In CBOW model, the words are projected by their previous and future words along the window size. This model is called bag-of-words because order of the words inside the window are not taken into consideration and the context is trained by neighboring information. Skip-gram model works similar to CBOW model with a few key differences. In skip-gram model words are used to project their neighboring words within the window

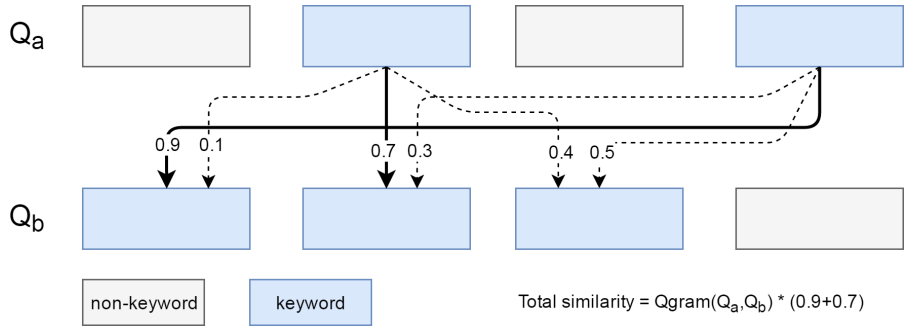

Figure 2: Procedure to calculate similarity between Q_a and Q_b.

instead of the other way around. And not all words within window are used but they are chosen randomly and they are weighted according to their proximity to the current word within the window.

Reviewing Word2Vec vectors trained with webchat dataset we have found some semantic relations between words, such as:

vec(*yillik*)+vec(*ucret*) = vec(*aidat*)

yillik means annual, *ucret* means fee in Turkish and *aidat* is another word in Turkish meaning periodic fee.

3.3.2 FastText

FastText method is a derivation of Word2Vec method but instead of focusing words as a whole it takes into account character n-grams within words (Bojanowski et al., 2016). Vectors are calculated for each character n-gram that appear in corpus and words are represented as the sum of vectors of character n-grams within the word. Authors argue that focusing on character n-grams instead of words yield better results for morphologically rich languages such as Turkish. Because some rare inflections of words may not appear in training corpus. However, since character n-grams are used to calculate a word vector, embeddings for those words can be used when they appear in other documents.

3.4 Extracting Keywords

Keywords play an important role for our question answering system since similarity between sentences is calculated by sole keywords. In order to extract keywords, we have clustered messages from Facebook messenger and Webchat datasets by using k-means algorithm with *k=100*. Before clustering process, messages are represented with 300 dimensional vectors, then dimensionality reduction(Maaten and Hinton, 2008) is applied into them. Each message is tokenized, then word vectors of tokens are averaged in order to obtain a representation of the message. In Equation 2, vector representation of j-th document is denoted in which n is the number of words in the document, w is a list of words.

Clustering results obtained from Facebook messenger have shown better performance since average length of messages in the dataset is less than the messages in Webchat dataset. Length of messages plays an important role in our clustering process, since clustered instances are vectors representing messages. Thefore, keywords in a message are not dominant for representing a vector of the message when the message becomes longer.We have selected 30 clusters with messages that are close to each cluster centroid. We have chosen topics of these 30 clusters as the scope of Q&A system. Then, for each of these important tokens are extracted according to their proximity to cluster centroid to be used as keywords. Moreover, keyword set is augmented by adding words with similar word vectors to these keywords. This process plays an important role since Turkish is a morphologically rich language. So, it becomes possible to cover same word with different suffixes without parsing words morphologically.

$$Doc_j = \frac{1}{n}\sum_{i=1}^{n} w_i \tag{2}$$

In Table 1, some sample questions from different topics that are found by clustering are shown with their English translations.

3.5 Question Answering

Given that Q_a and Q_b are two questions such that we want to calculate how much Q_a is similar to Q_b. Thus, we have developed a keyword based similarity metric that is a combination of cosine and q-gram similarity metrics. There are other studies that our approach shares some similarities, namely using keywords in an information retrieval based question answer system (Tirpude and Alvi, 2015; Kamdi and Agrawal,

2015). Our method mainly differs from these approaches by using word embeddings to extend keywords to include inlections and synonyms of keywords.

In Figure 2, the procedure utilized in our question answering system has been shown where Q_a and Q_b are two questions which are tokenized. Also, tokens which are keyword and non-keyword have been denoted. Similarity between two keywords are calculated. Each keyword in Q_a is matched with only one in Q_b such that both have the best similarity. The similarity between two keywords are calculated by multiplying their cosine and q-gram similarity values. Calculation of the cosine similarity is denoted in Equation 3 in which A and B are two n-dimensional vectors. Word embeddings trained with FastText and Word2Vec are used represantations of word for cosine similarity calculations.

$$Cosine - similarity = \frac{A \cdot B}{||A||\ ||B||} \tag{3}$$

In order to find the similarity between Q_a and Q_b, similarity values calculated for each token in Q_a is added up. Then, total similarity is multiplied by q-gram value of two questions.

```
function GetSimilarity(Tokens1, Tokens2)
  totalSimilarity = 0
  for( token1 in Tokens1)
     maxi = 0
     for( token2 in Tokens2)
          if(token1 in Model and token2 in Model)
               similarity = Cosine(token1,token2)
               similarity *= Qgram(token1,token2)
     if(maxi < similarity)
          maxi = similarity
     totalSimilarity += maxi
  return totalSimilarity
```

Question-answering in a banking domain is a different task when it is compared to a general domain. Banking domain has its own vocabulary. Therefore, a keyword based similarity metric shows a promising result.

4 EXPERIMENTAL RESULTS

In this section, we explain the experiments performed to show success of our keyword-based similarity measure. In order to perform tests, a set of 138 questions created that are various rephrases of questions from Q&A pair set. Experiments have been performed in two ways. Firstly, accuracy of finding different answers have been evaluated. In second part of experiments, accuracy for reliable answers is calculated. A reliable answer is the Q&A system admitting not being able to reply to the query instead of returning an unrelated answer.

The similarity metrics that are being compared are: Q-gram similarity, Word Mover's Distance (WMD), QA-wo-keyword is the proposed method without utilizing keywords, QA is the proposed method with keywords used for both user queries dataset questions, QA-q-keyword is a variant of QA algorithm where it utilizes the keyword set for questions obtained from Q&A pairs dataset. Also Word2Vec and FastText word embeddings are evaluated for each method separately except for Q-gram similarity which doesn't utilize word embeddings.

In the table, the best performing approach is QA-q-keyword which is based on word2vec model. It is a variant of the QA algorithm mentioned in Section 3.5. It only considers keywords in questions since a question asked by a customer may contain some words which are not in the keyword set. Yet, these words can show similarity with keywords. Because of the reason of that non-keywords are not ignored, more questions can be covered. On the other hand, Word Mover's Distance method degrades the performance.

When word embedding models are compared with each other, methods using FastText demonstrate slightly better performances except the QA-q-keyword method. Furthermore, we have also tested our approach for reliable answers since we do not want our system to return wrong answers. Thus, if it does not find a similar question, then it could not answer. In Table 3, responses which QA algorithm does not fail are considered as correct.

5 CONCLUSIONS

In this study, we have proposed a new way to compute similarity between two sentences by considering a set of keywords. This method has been mainly designed to be used in a retrieval based Turkish Q&A system. By utilizing keywords, the important words of queries and dataset are matched and performance has been increased significantly compared to methods that doesn't utilize keywords as our experiments have shown. In order to deal with challenges of a morphologically rich language such as Turkish word embeddings are used to expand keywords to include various inflections of words and calculate semantic similarity between words.

Experiments show that our hybrid keyword-based performed better than other similarity metrics. Mainly because this approach extends keywords by using word embeddings so different word inflections and misspelled words can be calculated as similar to keywords. Also, Q-gram similarity, a character based similarity is used to also increase scores even when

Table 1: Sample questions of selected topics.

Topic	Questions
IBAN	TR: IBAN numaram nedir? EN: What is my IBAN number?
	TR: IBAN numaramı öğrenmek istiyorum. EN: I want to learn my IBAN number.
Account Balance	TR: Hesabımda kaç tl var? EN: How much TL are there in my account?
	TR: Bakiyemi söyler misiniz? EN: Would you tell my balance?
Mobile App Problems	TR: Mobil uygulamaya giriş yapamıyorum. EN: I cannot login to mobile application.
	TR: Mobil cihazımdan bankacılık işlemlerini yapamıyorum. EN: I cannot perform banking operations via my mobile device.

Table 2: Accuracy results for correct replies.

Method	Word Embedding	Accuracy(%)
Q-gram	N/A	72.46
WMD	word2vec	44.20
QA-wo-keyword	word2vec	69.56
QA	word2vec	86.23
QA-q-keyword	**word2vec**	**90.57**
WMD	fastText	62.31
QA-wo-keyword	fastText	69.56
QA	fastText	86.95
QA-q-keyword	fastText	89.13

Table 3: Accuracy results for reliable replies.

Method	Word Embedding	Accuracy(%)
Q-gram	N/A	81.88
WMD	word2vec	87.68
QA-wo-keyword	word2vec	69.56
QA	**word2vec**	**92.75**
QA-q-keyword	**word2vec**	**92.75**
WMD	fastText	84.78
QA-wo-keyword	fastText	69.56
QA	**fastText**	**92.75**
QA-q-keyword	fastText	89.85

words are not exact matches.

For a future work, the process of extracting keywords from a set of messages can be automatized such that it becomes possible to import a new type of questions which contain a new possible list of keywords. Furthermore, we have manually determined the top 30 categories in question-answering system. This can also be automatized for adding new categories without human interaction.

ACKNOWLEDGEMENTS

The authors would like to thank Nihan Karslıoğlu for her invaluable contributions to the system and Erşah Yiğit Karademir for annotations. This work is supported by TUBITAK 3160116.

REFERENCES

Akın, A. A. and Akın, M. D. (2007). Zemberek, an open source nlp framework for turkic languages. *Structure*, 10:1–5.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Çelebi, E., Günel, B., and Şen, B. (2011). Automatic question answering for turkish with pattern parsing. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, pages 389–393. IEEE.

Kamdi, R. P. and Agrawal, A. J. (2015). Keywords based closed domain question answering system for indian penal code sections and indian amendment laws. *International Journal of Intelligent Systems and Applications*, 7(12):54.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.

Sen, M. U. and Erdogan, H. (2014). Türkçe için kelime temsillerinin ögrenimi learning word representations for turkish.

Soğancıoğlu, G., Çekiç, T., Köroğlu, B., Basmacı, M., and Ağın, O. (2017). Dialog management for credit card selling via finite state machine using sentiment classification in turkish language.

Soğancıoğlu, G., Köroğlu, B., and Ağın, O. (2016). Multi-label topic classification of turkish sentences using cascaded approach for dialog management system.

Tirpude, S. C. and Alvi, A. (2015). Closed domain keyword-based question answering system for legal documents of ipc sections and indian laws. *International Journal of Innovative Research in Computer and Communication Engineering*, 3(6):5299–5311.

Ukkonen, E. (1992). Approximate string-matching with q-grams and maximal matches. *Theoretical computer science*, 92(1):191–211.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Yurekli, B., Arslan, A., Senel, H. G., and Yilmazel, O. Turkish question answering.