Dialog-based Help Desk through Automated Question Answering and Intent Detection

Antonio Uva[‡], **Pierluigi Roberti**[†], **Alessandro Moschitti**[‡]
[‡]DISI, University of Trento, Italy
[†]Im Service Lab Srl, Italy

Abstract

Modern personal assistants require to access unstructured information in order to successfully fulfill user requests. In this paper, we have studied the use of two machine learning components to design personal assistants: intent classification, to understand the user request, and answer sentence selection, to carry out question answering from unstructured text. The evaluation results derived on five different real-world datasets, associated with different companies, show high accuracy for both tasks. This suggests that modern QA and dialog technology is effective for real-world tasks.

I moderni personal assistant richiedono di accedere ad informazioni non strutturate per soddisfare con successo le richieste degli utenti. In questo articolo, abbiamo studiato l'uso dell' apprendimento automatico per progettare due componenti di un personal assistant: classificazione degli intenti, per comprendere la richiesta dell'utente, e la selezione della frase di risposta per rispondere alle domande con testo non strutturato. I risultati della valutazione derivati da cinque diversi datasets del mondo reale, associati a diverse società, mostrano un'elevata precisione per entrambi i modelli. Ciò suggerisce che la moderna tecnologia di question answering *e dialogo è efficace per attività reali.*

1 Introduction

Help-desk applications use Machine Learning to classify user's request into intents. The information owned by companies generally is in free text form, from company's documents or websites. For example, corporate knowledge is typically encoded within documents in an unstructured way. This poses limitations on the effectiveness of standard information access. For example, searching documents by keywords is not a viable solution for the users, as they seldom can find an answer to their questions. The possibility of using QA systems to search for information on a corpus of documents, also through a dialogue system, offers an attractive solution for extracting the best information from the company knowledge bases.

IMSL company offers virtual agents that can be retrained based on the customer needs. The agent is composed of many Natural Language Understanding components, such as classifiers that map each user utterance in input to their corresponding intent. However, since it is not possible to forecast all the intents corresponding to the questions that the user are going to ask – which are potentially infinite – it is of paramount importance to have an automated QA system able to automatically provide the best answer (paragraph) extracted from a company owned knowledge base.

Information access is becoming an increasingly critical issue. Traditional Information Retrieval systems, used in industry, help the user in accessing information, but are often imprecise and impractical. Current search engines are an example of this. Searching for information on the web often requires a double effort for the user: first it is necessary to understand how to formulate a query in the most effective manner, and then filter out the proposed results in order to find the most relevant information.

In this paper, we described our QA system based on answer sentence selection and intent detection, and how we integrate them in a Conversational agent.

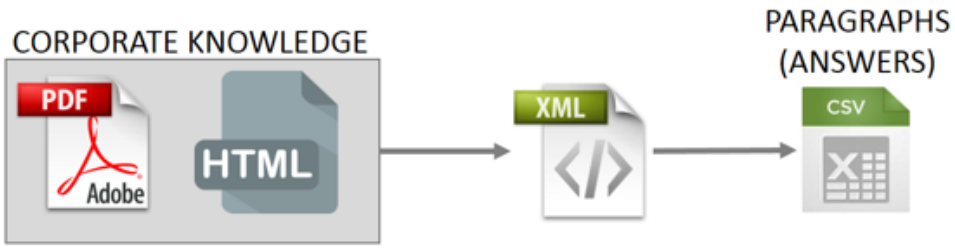

Figure 1: Paragraphs extraction.

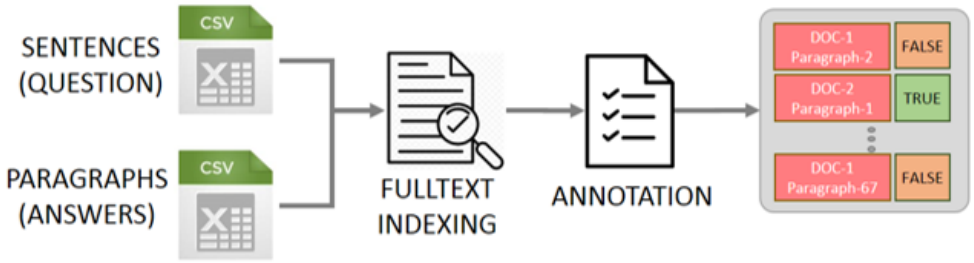

Figure 2: Annotation data for training.

2 Related Work

As today, the largest part of general-purpose QA services are provided by big tech companies such as Amazon Alexa, Google Home, Ask Yahoo!, Quora and many others. Unfortunately, these types of applications are not easily accessible for smaller companies, as the offered QA service cannot be easily adapted to handle corporate knowledge, which is in form of unstructured text. To build their own solutions SMEs can exploit QA components such as Answer Sentence Selection.

In recent years, deep learning approaches have been successfully applied for automatically modeling text pairs, e.g., (Lu and Li, 2013; Yu et al., 2014). Additionally, a number of deep learning models have been recently applied to QA, e.g., Yih et al. (2013) applied CNNs to open-domain QA; Bordes et al. (2014) propose a neural embedding model combined with the knowledge base for open-domain QA. Iyyer et al. (2014) applied recursive neural networks to factoid QA over paragraphs. Miao et al. (2016) proposed a neural variational inference model and a Long-short Term Memory network for the same task. Yin et al. (2016) proposed a siamese convolutional network for matching sentences that employ an attentive average pooling mechanism, obtaining state-of-the-art results in various tasks and datasets.

The work closest to this paper is by Yu et al. (2014) and Severyn and Moschitti (2015). The former presented a CNN architecture for answer sentence selection that uses bigram convolution and average pooling, whereas the latter use convolution with k-max pooling.

Nowadays, supporting customers in their activities across applications and websites is becoming always more demanding, due a large number of customers and the variety of topics that have to be covered.

New tools, such as chatbots, able to answer frequently asked questions, i.e., FAQs, are rising in response to this needs. Classifying the user need expressed in a natural question, into a predefined set of categories, allow conversational agents to recognize which users are asking which types of questions and to react accordingly.

Traditional approaches to this problem include the use supervised approaches such as Support Vector Machines (SVM) (Cortes and Vapnik, 1995), Boosting (Iyer et al., 2000; Schapire and Singer, 2000), Kernel machines operating on input structured objects (Moschitti, 2006; Lodhi et al., 2002) and Maximum Entropy models (Yaman et al., 2008).

In the latest years, new models such as Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) (Cortes and Vapnik, 1995), Gated Recurrent Unis (GRU) (Chung et al., 2014) and Convolutional Neural Networks (CNN) (Lecun et al., 1998; Kim, 2014) were established as state-of-the-art ap-

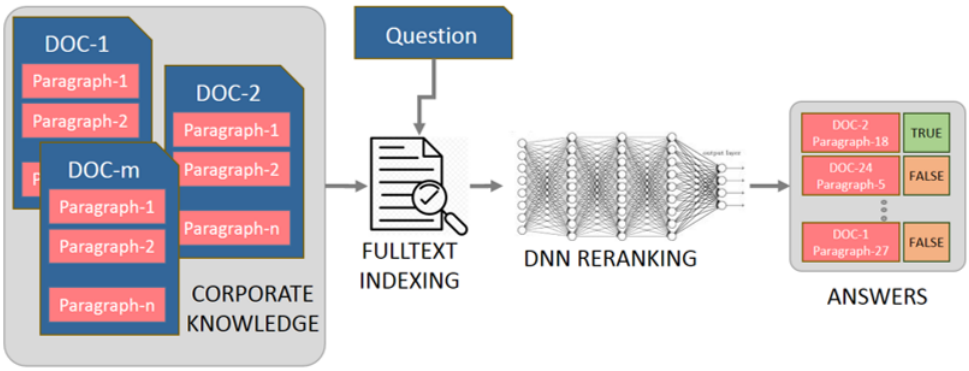

Figure 3: system architecture.

proaches for text classification.

3 System Description

Our QA system allows for extracting portions of text from company documents or from websites. This information is then organized into paragraphs, which are then used to provide an answer to the user's questions.

One practical problem is the fact that not all PDF files encode text, and many fail to preserve the logical order of the text. Thus, in order to extract paragraphs, we used pdf2text.

Another practical problem we need to solve was to keep portions of text separated by punctuation together: such as bullet lists or very structured paragraphs. Our designed tool automatically assigns a reference index or summary to each paragraph to improve subsequent searches (see Figure 1).

Subsequently, each question and answer pair must be annotated with correctness (label TRUE/FALSE). This allows us to create a training set to train the re-ranking network (see Figure 2).

The final system, shown in Figure 3, therefore allows for using the target company data, appropriately reorganized into paragraphs, to provide answers to the user's request. On average we provide from 3 to 5 answers for each question. However, we also provide the reference to the document and the summary which the paragraph refers to.

4 Answer Sentence Selection (AS2)

The AS2 goal is to rank a list of answer candidates by their similarity with respect to an input question q_i. We design a network that includes relational information between questions and answers. Our results show that CNNs reach better performance than traditional IR models based on bag of words.

4.1 Model

The architecture of the network used for mapping sentences in embedding vectors is showed in Figure 4 and is inspired to the CNNs employed by Severyn and Moschitti (2015) to perform many classification activities over sentences. It includes two main components:

(i) an encoder that map an input document s_i into a vector x_{s_i} and (ii) a feed-forward network that computes the similarity between input sentences.

Our network takes two sentences in input, i.e., a question and a text paragraph that may contain an answer, and it represents each of them into vectors of fixed-size dimension $x_s \in \mathbb{R}^m$.

The sentence model is composed of a sequence of convolutional maps followed by some pooling operations. Such model achieves the state of the art in many NLP tasks (Kalchbrenner et al., 2014; Kim, 2014).

Then, the sentence vectors, x_{s_i} corresponding to the questions and answers, are concatenated together and passed to the following neural network layers. These are composed of a non-linear hidden layer and an output layer with a sigmoid activation unit. At the end, the network returns a value between 0 and 1 corresponding to the relevancy of the answer with respect to the question.

Finally, we included word overlap embeddings encoding relational information between words in questions and answers (Severyn and Moschitti, 2016).

5 Intent Classification

We adopted advanced techniques, such as deep learning models, to classify the user need, which is

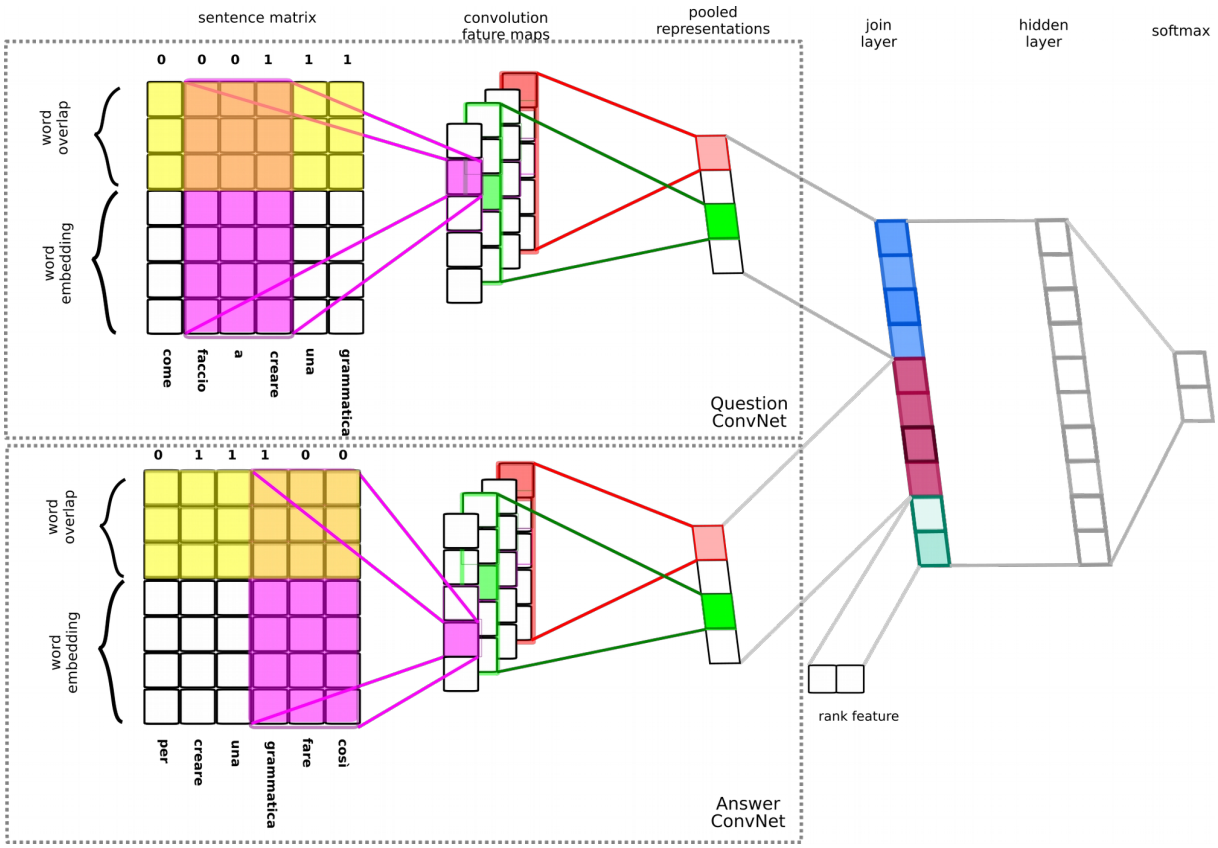

Figure 4: Architecture of the network computing relevancy of answers with respect to the questions. The network is composed by two subnetworks: (1) the Question ConvNet that encodes input questions into a fixed-size vector and (2) the Answer ConvNet that encodes the answer into a fixed-size vector. The vectors of questions and answers are concatenated into a new vector (join layer), where a new embedding is added, which embeds rank information. Then, a Multilayer perceptron (MLP) composed of an hidden layer and a softmax classifier, returns a value between 0 e 1. This indicates the relevancy of an answer with respect to a question.

semantically expressed by the user question, into a predefined set of categories, i.e., intents.

We used some common deep learning models for solving the intent detection task. The main point of our study is to test those models and observe how they perform on datasets containing real user questions addressed to a virtual agent, operating in the banking/financial sector.

At this stage, we dot not consider novel methods based on transformer architecture such as BERT (Devlin et al., 2019), which require a a large amount of resources, typically not avialable to SMEs. Instead, we focused on lighter approaches that can run on small GPUs. We report our experiments and discuss the obtained results using such lighter models.

5.1 Models

SVM (baseline) fed with word features, derived from the text of the utterances.

LSTM using recurrent units that take in input the embedding x_t of the current word at time step t and the hidden vector encoding the sub-phrase at previous step, i.e., h_{t-1}, and return the vector representation of the phrase at step h_t

CNN uses a set of convolutional filters of different size and max pooling operations to extract the most important features, e.g., bigrms, trigams, etc. . . , which represent the sentence meaning.

LSTM + CNN based on an architecture composed of two layers: an LSTM layer that builds a fixed-size vector representation of the sentence at each word, and a convolutional layer. The latter applies a set of convolutional operations on the

	DEV. SET			TEST SET		
Models	MAP	MRR	P@1	MAP	MRR	P@1
BM25	64.20 ± 0.00	**70.20 ± 0.00**	**57.60 ± 0.00**	55.40 ± 0.00	62.40 ± 0.00	46.70 ± 0.00
CNN	**65.04 ± 1.10**	69.34 ± 1.36	53.34 ± 2.66	**68.38 ± 1.08**	**72.21 ± 1.33**	**57.42 ± 2.16**

Table 1: The results of the QA model on the dev. and test set of IMSL-WIKI corpus

representations returned by the first layer.

CNN + CNN composed of two CNN layers, where the second layer takes the previous layer representation as input, and applies a set of convolutional filters and pooling operation to compute the final vector representation of the sentence.

6 Experiments

In this section, we first describe the datasets we used in our experiments, then we provide the results on the answer sentence selection and the intent classification tasks. Finally, we report an end-to-end evaluation of our system.

6.1 Data Description

We built our datasets by collecting samples of questions asked by users to conversational agent for either **Credit Institution** or **Bank** websites. We collected two intent corpora from each data provider, resulting in a total of four datasets.

Istituto Credito - synthetic (IC$_s$): This corpus was created by expert dialog engineers. It contains a set of utterances annotated with their corresponding intents. The subject of questions are diverse and spans over many topics. For example, some questions seek information over the bank branch locations, problems regarding how to cash checks, and requests of availability of finance products. It contains 2,305 training examples, and 593 test examples, for a total of 2,898 examples.

Istituto Credito - full (IC$_f$): This dataset is composed of synthetic questions, generated by language engineers. Subsequently, it has been augmented to take into account also real sentences, retrieved from website chat-bot of a well known Credit Institution operating in Italy. It contains 2,898 training examples and 770 test examples, for a total of 3,668 examples.

Banca - Area Informativa (Banca$_{AI}$): This dataset contains real questions asked by users about the Area Informativa of a bank. It includes 3,947 training examples, and 987 test examples, for a total of 4,934 examples divided in 282 intents.

Banca - Internet Banking (Banca$_{IB}$): This dataset includes real questions asked by users about the iBanking service offered from a well known Italian bank. It includes 4,380 training instances and 1,906 test instances divided in 251 intents.

Answer Sentence Selection data: We used an in-house dataset called IMSL-WIKI, which contains a list of question and answer regarding some of the products and services sold by IM Service Lab. For each question, a paragraph list was collected using an off-the-shelf search engine, i.e., Lucene, and manually annotated as either relevant or irrelevant. The dataset is divided into two parts, i.e., a training and test sets, which contain a total of 5,190 and 1,240 QA pairs, respectively. For each question, we retrieved a list of 10 candidate answers.

6.2 Model results

In this section we report the performance of our two main machine learning components of our system: Answer Sentence Selection and Intent Classification.

6.2.1 Answer Sentence Reranking

Table 1 reports the performance of the neural network and the baseline system. The first row, i.e., BM25, shows the baseline system, while the second row shows the performance of the CNN. The systems are evaluated according to the Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Precision at 1 (P@1). The final results reported at the bottom is obtained as the average of 5 different models trained and evaluated on the test set. For each measure in the table, we report both mean and standard deviation computed on dev. and test sets.

We used a small fraction of the training set, i.e., 15% of the data, for early stopping. As it can be seen from the table, CNN performs about 1 point more than the baseline algorithm (BM25) in terms of MAP on the dev. set, and almost 10 absolute points more of MAP on the test set.

In addition, we observe an increase of 9.8 absolute points in terms of MRR, and 10.65 absolute points of P@1 on the test set. The difference between results on dev. and test sets can be explained

Models	Test Sets (Accuracy) IC_s	IC_f	$Banca_{AI}$	$Banca_{IB}$
Baseline (SVM)	0.7622	0.8065	0.8197	0.7235
CNN	0.7718	0.8058	0.8241	0.7633
CNN + CNN	0.7577	0.8094	0.8328	0.7663
LSTM	0.7698	0.8131	**0.8529**	**0.7843**
LSTM + CNN	**0.7737**	**0.8231**	0.8224	0.7479

Table 2: Accuracy over the datasets.

Models	Test Sets (F_1 score) IC_s	IC_f	$Banca_{AI}$	$Banca_{IB}$
Baseline (SVM)	0.7595	0.8151	0.8009	0.7108
CNN	0.7722	0.8078	0.8064	0.7476
CNN + CNN	0.7606	0.8117	0.8158	0.7499
LSTM	0.7689	0.8163	**0.8386**	**0.7691**
LSTM + CNN	**0.7742**	**0.8252**	0.8065	0.7344

Table 3: F_1 score over the datasets.

by the fact that the used dev. set is very small: only 124 list of questions and 1,239 Q/A pairs, which made it difficult to optimize the three ranking metrics at the same time, so we focused on MAP.

6.2.2 Intent Classification

We ran state-of-the-art neural classifiers described in Section 6.2.1 on Credit Institute and Bank datasets. To choose the best performance, we used 30% of training data as validation set and select the best hyperparameters. We compare the performance of neural models with respect to strong baseline classifiers, i.e., SVMs, and report the results in terms of Accuracy (Table 2) and F_1 (Table 3). The tables show that the final performance heavily depends on the used dataset and models.

Istituto Credito (IC) datasets. Regarding the IC synthetic dataset, the best model, i.e., LSTM+CNN, obtains Accuracy of 77.37 and a micro-avg F_1 of 0.7742. This is about one absolute point of Accuracy higher than the base SVM model (77.37 vs. 76.22) and 1.47 absolute points of F_1 more than the base model (77.42 vs. 75.95). Similarly, on the IC full dataset, the performance of the best model, i.e, LSTM+CNN, achieved an accuracy of 82.31%, which is 0.66% absolute points better than the base model (82.31 vs. 80.65) and a micro-avg F_1 of 82.52, which is about one point better than the base SVM model (82.52 vs 81.51).

Banca datasets. Regarding Banca AI dataset, the best model, i.e., LSTM obtained accuracy of 85.29, which is about 4 absolute points better than the base SVM model (85.29 vs. 81.97). Also, in terms of F_1, the best model obtained 3.77 absolute points more than the baseline (82.86 vs 80.09). Regarding the Banca IB dataset, the best model, i.e., LSTM, obtained around 6 points more both in terms of Accuracy (78.43 vs 72.35) and F_1 (76.91 vs 71.08).

6.3 End-to-End system evaluation

We trained and evaluated our system using samples of data collecting from IMSL customers.

We noted that the accuracy of the system improved because more answers are generally provided (from 3 to 5) to the user's question, thus allowing to almost certainly provide the correct answer.

The only point of attention is the fact that there is not always a valid answer to the user's request in company knowledge. Indeed, the questions related to the user's personal profile or data cannot be precisely answered by the company documentation.

Furthermore, it often happens that the company policy prevents to provide explicit answers to specific user problems. In all these cases, it is therefore necessary to support the QA system with operators, who can provide personal answers or those not coded in the corporate knowledge.

7 Conclusions

In this paper, we have presented a modern dialog system for real-world applications. We have tested advanced technology for QA and intent classification on several datasets derived from company data, such as Banks and Credit Institutions. The results show a promising direction for SMEs to build their own effective access to unstructured data.

References

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620, Doha, Qatar, October. Association for Computational Linguistics.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. In *Machine Learning*, pages 273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Raj D Iyer, David D Lewis, Robert E Schapire, Yoram Singer, and Amit Singhal. 2000. Boosting for document routing. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 70–77.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 633–644.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *Advances in neural information processing systems*, pages 1367–1375.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *11th conference of the European Chapter of the Association for Computational Linguistics*.

Robert E Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 373–382.

Aliaksei Severyn and Alessandro Moschitti. 2016. Modeling relational information in question-answer pairs with convolutional neural networks. *arXiv preprint arXiv:1604.01178*.

Sibel Yaman, Li Deng, Dong Yu, Ye-Yi Wang, and Alex Acero. 2008. An integrative and discriminative technique for spoken utterance classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1207–1214.

Scott Wen-tau Yih, Ming-Wei Chang, Chris Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*.