SuperAgent: A Customer Service Chatbot for E-commerce Websites

Lei Cui[*], **Shaohan Huang**[*], **Furu Wei**, **Chuanqi Tan**, **Chaoqun Duan**, and **Ming Zhou**

Microsoft Research Asia
{lecu,shaohanh,fuwei,v-chutan,v-chadu,mingzhou}@microsoft.com

Abstract

Conventional customer service chatbots are usually based on human dialogue, yet significant issues in terms of data scale and privacy. In this paper, we present SuperAgent, a customer service chatbot that leverages large-scale and publicly available e-commerce data. Distinct from existing counterparts, SuperAgent takes advantage of data from in-page product descriptions as well as user-generated content from e-commerce websites, which is more practical and cost-effective when answering repetitive questions, freeing up human support staff to answer much higher value questions. We demonstrate SuperAgent as an add-on extension to mainstream web browsers and show its usefulness to user's online shopping experience.

1 Introduction

Customer service plays an important role in an organization's ability to generate income and revenue. It is often the most resource-intensive department within a company, consuming billions of dollars a year to change the entire perception customers hold. Support staff spend a lot of time answering questions via telephone or messaging applications to make sure are customers satisfied with their business. This traditional customer service has two problems: First, staff usually receive repetitive questions asked by a variety of customers, which can be cost-effectively answered by machines. Second, it is difficult to support 7×24 services, especially for most non-global businesses. Therefore, chatbots can be a great way to supplement customer service offerings since they are more economical and indefatigable, and free up support staff to answer much higher value queries.

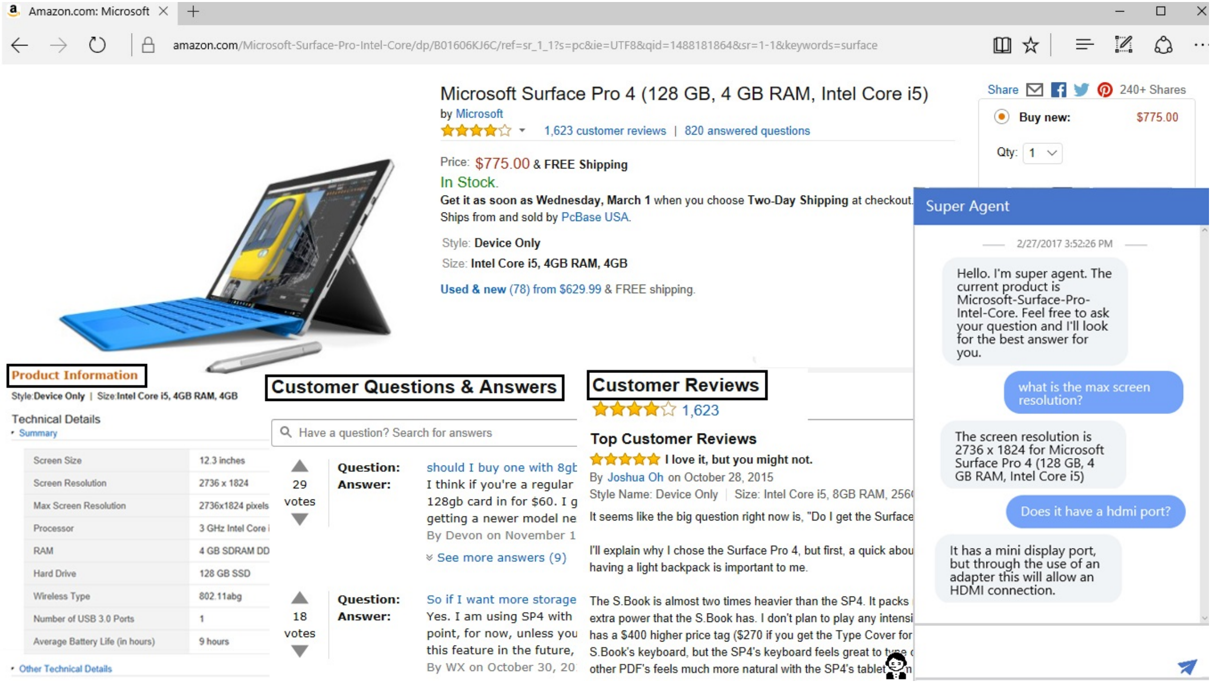
Recently, virtual assistants for customer service have become more and more popular with customer oriented businesses. Most of them are built from human conversations in the past, which is straightforward but faced with problems of data scale and privacy. Most of the time, customers need to wait online to get a support staff person's answer, which is less effective and difficult to scale up. Meanwhile, customers may have privacy concerns about the conversations, hence conversations with customers cannot be easily leveraged to train a chatbot. It is essential to find large-scale and publicly available customer service data sources on which to build such assistants.

In this paper, we demonstrate SuperAgent, a powerful customer service chatbot leveraging large-scale and publicly available e-commerce data. Nowadays, large e-commerce websites contain a great deal of in-page product descriptions as well as user-generated content, such as Amazon.com, Ebay.com, and JD.com. Figure 1 shows a product page from Amazon.com, which contains detailed Product Information (PI), a set of existing customer Questions & Answers (QA), as well as sufficient Customer Reviews (CR). This crowd-sourcing style of data provides appropriate information to feed into chat engines, accompanying human support staff to deliver better customer service experience when online shopping.

We define the problem as follows: given a specific product page and a customer question, SuperAgent selects the best answer from existing data sources within the page (PI+QA+CR). If it cannot find the answer, it indicates that no replies will be generated. Specifically, we decompose the chat engine into three sub-engines: 1) a fact question answering engine for PI; 2) an FAQ search engine for QA; 3) an opinion mining & text question an-

[*] The first two authors contribute equally to this work.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations, pages 97–102
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-4017



Amazon.com: Microsoft
amazon.com/Microsoft-Surface-Pro-Intel-Core/dp/B01606KJ6C/ref=sr_1_1?s=pc&ie=UTF8&qid=1488181864&sr=1-1&keywords=surface
Share
240+ Shares

Microsoft Surface Pro 4 (128 GB, 4 GB RAM, Intel Core i5)
by Microsoft
★★★★☆ 1,623 customer reviews | 820 answered questions

Price: $775.00 & FREE Shipping
In Stock.
Get it as soon as Wednesday, March 1 when you choose Two-Day Shipping at checkout.
Ships from and sold by PcBase USA.

Style: Device Only
Size: Intel Core i5, 4GB RAM, 4GB

Used & new (78) from $629.99 & FREE shipping.

● Buy new: $775.00
Qty: 1

Product Information

Style:Device Only | Size:Intel Core i5, 4GB RAM, 4GB

Technical Details
- Summary

Screen Size	12.3 inches
Screen Resolution	2736 x 1824
Max Screen Resolution	2736x1824 pixels
Processor	3 GHz Intel Core i
RAM	4 GB SDRAM DD
Hard Drive	128 GB SSD
Wireless Type	802.11abg
Number of USB 3.0 Ports	1
Average Battery Life (in hours)	9 hours

- Other Technical Details

Customer Questions & Answers

Have a question? Search for answers

29 votes
Question: should I buy one with 8gt
Answer: I think if you're a regular 128gb card in for $60. I g getting a newer model ne
By Devon on November 1
See more answers (9)

18 votes
Question: So if I want more storage
Answer: Yes. I am using SP4 with point, for now, unless you this feature in the future,
By WX on October 30, 20

Customer Reviews

★★★★☆ 1,623

Top Customer Reviews

★★★★★ I love it, but you might not.
By Joshua Oh on October 28, 2015
Style Name: Device Only | Size: Intel Core i5, 8GB RAM, 256

It seems like the big question right now is, "Do I get the Surface

I'll explain why I chose the Surface Pro 4, but first, a quick abou having a light backpack is important to me.

The S.Book is almost two times heavier than the SP4. It packs extra power that the S.Book has. I don't plan to play any intensi has a $400 higher price tag ($270 if you get the Type Cover for S.Book's keyboard, but the SP4's keyboard feels great to type other PDF's feels much more natural with the SP4's tablet n

Super Agent

2/27/2017 3:52:26 PM

Hello. I'm super agent. The current product is Microsoft-Surface-Pro-Intel-Core. Feel free to ask your question and I'll look for the best answer for you.

what is the max screen resolution?

The screen resolution is 2736 x 1824 for Microsoft Surface Pro 4 (128 GB, 4 GB RAM, Intel Core i5)

Does it have a hdmi port?

It has a mini display port, but through the use of an adapter this will allow an HDMI connection.

Figure 1: An example of a product page from Amazon.com, including product information, customer Q&A, and reviews. SuperAgent is an add-on extension located at the bottom right corner.

swering engine for CR. In addition, we also add a chit-chat engine as the back-fill to make conversations as smooth as possible. To improve the overall online shopping experience, we demonstrate SuperAgent as an add-on extension to mainstream web browsers such as Microsoft Edge and Google Chrome, where the conversation UI is shown at the bottom right corner, as shown in Figure 1.

Compared to conventional customer service chatbots, SuperAgent has several promising advantages:

1. SuperAgent can easily leverage crowd-sourcing styles, as well as large-scale and publicly available e-commerce data,

2. SuperAgent contains a set of state-of-the-art NLP and machine learning techniques, including fact QA, FAQ search, opinion mining, text QA, and chit-chat conversation.

3. SuperAgent is integrated into e-commerce websites as an add-on extension, which can directly improve customer's online shopping experience.

The rest of the paper is organized as follows: The system details are described in Section 2. The usability analysis is presented in Section 3. Section 4 introduces some related work. Section 5 concludes the paper and suggests future directions.

2 System Details

2.1 Overview

Figure 2 shows the system overview of SuperAgent. As the figure shows, when the product page is first visited, SuperAgent crawls the html information and scrape PI+QA+CR data from the webpage. The advantage of this design pattern is that we do not need to deploy web crawlers for the websites. Instead, when users visit the page, SuperAgent will be notified since the add-on extension is associated with each webpage. Therefore, SuperAgent brings very few additional web loads to the host websites. Besides, this architecture makes data updates very easy to implement, where frequently-visited pages get updated frequently and vice versa. After that, given an input query from a customer, different engines are processed in parallel. If one of the answers from the first three engines has high confidence, the chatbot returns with the answer as the response. Otherwise, the chit-chat engine will generate a reply from the predefined permitted response sets. Next, we introduce these engines in detail.

2.2 Fact QA for Product Information

The fact QA engine is designed for answering questions regarding the facts of the product. The product information is stored in the

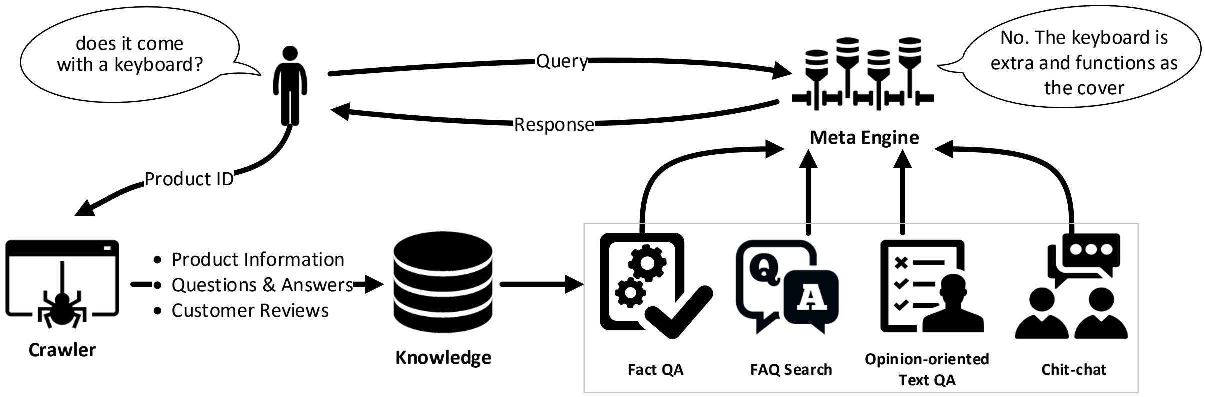


Figure 2: The system overview of SuperAgent

format of knowledge triples ⟨#, attribute_name, attribute_value⟩, where # represents the product name in our scenario. We focus on simple fact inquiry in this paper. As a result, the task is transformed to matching the question to the attribute names in the product information. This is achieved with a deep learning based matching framework. Specifically, the input question is matched with every attribute name using the DSSM model (Huang et al., 2013). We select the top-ranked attribute name which passes a predefined threshold as the result from attribute matching. The corresponding attribute value is further used to generate the response sentences with predefined templates. The engine's output is shown as follows:

> Q: What's the CPU?
>
> A: The processor is 3 GHz Intel Core i5 for Microsoft Surface Pro 4 (128 GB, 4 GB RAM, Intel Core i5)
>
> Q: What is the pre-installed operating system?
>
> A: The operating system is Windows 10 Pro for Microsoft Surface Pro 4 (128 GB, 4 GB RAM, Intel Core i5)

2.3 FAQ Search for Customer QA Pairs

The FAQ search engine is defined as follows: given a set of QA pairs $P = \{q_i, a_i\}_{i=1}^{n}$ and a customer's question q, we find the most similar q_j in P and return the corresponding a_j as the reply. For example, given two QA pairs:

> Q: does it come with a keyboard
>
> A: No. The keyboard is extra and functions as the cover
>
> Q: does it come with the pen
>
> A: yes it does

Then, if the user asks "does it have a keyboard", the question "does it come with a keyboard" will be matched and the answer should be "No. The keyboard is extra and functions as the cover". Therefore, the question ranking problem is essential for an FAQ search engine.

Formally, given two questions q and $q\prime$, the question ranker will learn a mapping function f where $f(q, q\prime) \rightarrow [0, 1]$, so f is actually a semantic similarity metric between two questions, indicating whether they convey the same meaning. We train a regression forest model (Meinshausen, 2006) and use the following features: monolingual word aligner (Sultan et al., 2014), DSSM model (Huang et al., 2013), word embedding composition (max, sum, idf-sum) with GloVe (Pennington et al., 2014), n-gram overlap, subsequence matching, PairingWords (Han et al., 2013), word mover's distance (Kusner et al., 2015).

The model is evaluated on the dataset for SemEval-2016 Task 1. The mean accuracy (Pearson Correlation) of our model on five datasets (0.78455) significantly outperforms the 1st place team (0.77807)[1]. Specifically, the accuracy on the question-to-question dataset is 0.75773, surpassing the 1st place team (0.68705) by a large margin. This confirms the effectiveness of our model on the FAQ search task.

2.4 Opinion-Oriented Text QA for Reviews

Customer reviews provide rich information for different aspects of the product from users' per-

[1] http://alt.qcri.org/semeval2016/task1/index.php?id=results

spective. They are very important resources for answering opinion-oriented questions. To this end, we first split the review text into sentences and run opinion mining modules to extract the aspects and corresponding opinions. We then use the text question answering module to generate responses (i.e. review sentences).

For opinion mining, we use a hybrid approach (Qiu et al., 2011) to extract the aspects from review sentences. We also run the sentiment classifier (Tang et al., 2014) to determine the polarity (i.e. positive, negative, and neutral) of the sentence regarding the specific aspect mentioned. The aspects and polarity are indexed together with keywords using the Lucene toolkit[2].

For text QA, given an input query, it outputs the answer based on the following three steps:

- *Candidate retrieval*, which searches the query by Lucene to get candidate sentences.
- *Candidate ranking*, which ranks all candidate sentences with a regression based ranking framework.
- *Candidate triggering*, which decides whether it is confident enough to output the candidate.

Specifically, for candidate retrieval, we use the default ranker in Lucene to retrieve the top 20 candidate sentences. For candidate ranking, we build a regression based framework to rank all candidate sentences based on features designed at different levels of granularity. Our feature set consists of WordCnt, translation model, type matching, WordNet, and two neural network based methods BiCNN (Yu et al., 2014) and MatchLSTM (Wang and Jiang, 2016). We conduct experiments on the WikiQA (Yang et al., 2015) dataset. The results show that we achieve state-of-the-art results with 0.7164 in terms of MAP and 0.7332 in terms of MRR. For candidate triggering, as the ranking model outputs a regression score for each candidate sentence, we only output the candidate sentence whose score is higher than the threshold selected on the development set. The engine's output examples are shown as follows:

> Q: what do you think of the battery life of surface pro 4?
>
> R: I've been getting ˜7-9 hours of battery life on the SP4 which is more than enough to get me through a school day.
>
> Q: Is the screen size of surface pro 4 appropriate for reading?
>
> R: The screen is not too small like the iPad, neither is it bulky like a laptop.

2.5 Chit-chat Conversation Modeling

The chit-chat engine is mainly designed to reply to greeting queries such as "hello" and "thank you", as well as queries that cannot be answered by the previous three engines, such as "you are so cute". However, general chit-chat engines tend to be topic-deviated so that the replies may be irrelevant. To avoid such deviations, we follow the smart reply approach for email reply suggestions (Kannan et al., 2016) to predefine a permitted response set. Formally, the chit-chat model is an attention-based LSTM seq2seq model (Bahdanau et al., 2014) trained on twitter conversation data (˜43 million query-reply pairs). We select the 5 million most frequent non-duplicate short replies (less than 15 words) as the permitted response set, most of which are greetings and common replies. The end-to-end perplexity of the seq2seq model is 16.9, which is similar to Kannan et al.'s result. The engine's output is very topic-coherent, which is shown as follows:

> Q: hello
>
> R: hey how are you?
>
> Q: thank you
>
> R: you're very welcome sir
>
> Q: you are so cute
>
> R: u r more

2.6 Meta Engine

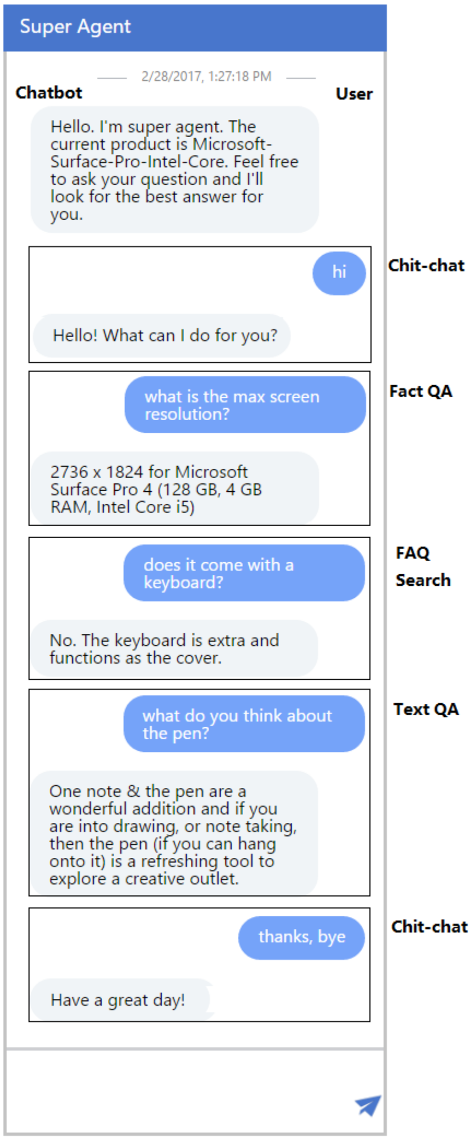
For each query, SuperAgent will call the above-mentioned sub-engines in parallel. The meta engine is then used to merge and prioritize the results from the different engines. We use a simple strategy to implement the meta engine, which prefers results from the engines in order of fact QA, FAQ search, text QA and chit-chat engine according to tunable threshold.

3 Usability Analysis

In this section, we discuss the reason why SuperAgent is necessary and how customers use it. When customers visit e-commerce websites, they often

[2] http://lucene.apache.org

need to know what other people feel after the purchase. In addition, customers are also interested in the answers to similar questions that other people have. We randomly select 670 product pages from Amazon.com and get a total of 29,471 customer QA pairs, almost 44 pairs per product on average. We also get 72,402 customer reviews, about 108 reviews per product on average. The number is much higher for a popular product, making it difficult for users to read all of them. For example, in Figure 1, there are more than 800 QA pairs existing within the page, which is impossible to go through by customers. Besides, there are also more than 1,200 high quality customer reviews in the page of Figure 1, which is overwhelming for users to read. Therefore, it is crucial to integrate these data into a unified knowledge base and provide easy access to all customers.


Super Agent

2/28/2017, 1:27:18 PM

Chatbot

User

Hello. I'm super agent. The current product is Microsoft-Surface-Pro-Intel-Core. Feel free to ask your question and I'll look for the best answer for you.

hi

Hello! What can I do for you?

what is the max screen resolution?

2736 x 1824 for Microsoft Surface Pro 4 (128 GB, 4 GB RAM, Intel Core i5)

does it come with a keyboard?

No. The keyboard is extra and functions as the cover.

what do you think about the pen?

One note & the pen are a wonderful addition and if you are into drawing, or note taking, then the pen (if you can hang onto it) is a refreshing tool to explore a creative outlet.

thanks, bye

Have a great day!

Figure 3: A case study of SuperAgent

Figure 3 shows a typical scenario when a customer asks SuperAgent for help. When the customer opens up the chat window within web browsers, SuperAgent first detects which product is being visited. SuperAgent then makes a self-introduction and confirms that the customer is visiting the product. Subsequently, customers may greet SuperAgent or ask specific questions. As Figure 3 shows, SuperAgent is able to answer fact questions using in-page product information, conduct an FAQ search from customer QA pairs, get text QA answers from customer reviews, and finally greet customers using the chit-chat engine. The dialogs are coordinated by the meta engine so that different queries go to corresponding engines respectively. Since e-commerce websites get updated frequently and fresh user-generated content emerges continuously, SuperAgent also updates the data and models periodically according to the frequency of customers' visits.

4 Related Work

Customer service chatbots can be roughly categorized into two types: first-party and third-party. First-party chatbots refer to conversation engines developed by large enterprises for their own business to improve customer service quality and reduce overall customer service budget. This often happens in consumer-driven industries such as banking, telecoms, and e-commerce. One example is the recently launched chatbot Erica from Bank of America, which helps customers with banking-related problems. Another example is the AT&T support chatbot that helps people answer FAQs related to different business questions.

Third-party chatbots refer to open source building blocks that help developers to build their conversation engines, such as Microsoft Bot Framework[3], Facebook Messenger[4], Google Assistant[5], and Amazon Lex[6]. These chatbot makers build and connect intelligent conversation engines to interact with customers naturally wherever they are. In addition, they are highly customizable in terms of real scenarios with third-party data.

It is worth mentioning that we position SuperAgent as a third-party chatbot because it utilizes

[3] https://dev.botframework.com/

[4] https://messengerplatform.fb.com/

[5] https://assistant.google.com/

[6] https://aws.amazon.com/lex/

publicly available third-party data. Moreover, it improves the overall online shopping experience for various products in an e-commerce website.

5 Conclusion and Future Work

We have developed SuperAgent, a customer service chatbot for e-commerce websites. Compared to conventional customer service chatbots, SuperAgent takes advantage of large-scale, publicly available, and crowd-sourced customer data. In addition, SuperAgent leverages state-of-the-art NLP and machine learning techniques, including fact QA, FAQ search, opinion-oriented text QA, as well as chit-chat conversation modeling. Usability analysis shows that SuperAgent has improved the end-to-end user experience in terms of online shopping. It is more convenient for customer's information acquisition especially when a product page contains too much user-generated content.

In the future, we will focus on two main problems. First, we need to integrate a customer's query intent detection module, so that we can better leverage individual engines. Second, we will have a deeper delve on multi-turn queries, where context modeling will be further investigated.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 44–52. http://www.aclweb.org/anthology/S13-1005.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. ACM, New York, NY, USA, CIKM '13, pages 2333–2338. https://doi.org/10.1145/2505515.2505665.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufman, Balint Miklos, Greg Corrado, Andrew Tomkins, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016).*. http://www.kdd.org/kdd2016/papers/files/Paper_1069.pdf.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, ICML'15, pages 957–966. http://dl.acm.org/citation.cfm?id=3045118.3045221.

Nicolai Meinshausen. 2006. Quantile regression forests. *J. Mach. Learn. Res.* 7:983–999. http://dl.acm.org/citation.cfm?id=1248547.1248582.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.* 37(1):9–27. https://doi.org/10.1162/coli_a_00034.

Md Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics* 2:219–230. https://transacl.org/ojs/index.php/tacl/article/view/292.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1555–1565. http://www.aclweb.org/anthology/P14-1146.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 2013–2018. https://doi.org/10.18653/v1/D15-1237.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *Proceedings of the Deep Learning and Representation Learning Workshop: NIPS* .