Answering Binary Questions on Products In A Noisy Setting

Guglielmo Menchetti, Lorenzo Norcini, Federico Sandrelli
University of Illinois at Chicago
Politecnico di Milano
{gmench2,lnorci2,fsandr2}@uic.edu

Abstract

Product reviews are ubiquitous in most e-commerce platforms and play an important role in aiding users in choosing items. Some of these platforms also offer Question and Answer (QA) systems to allow users to ask product-related inquiries to the community. Often, it is the case that a product is flooded with hundreds of reviews, making it difficult for the inquirer to browse through all of them. Under the reasonable assumption that most questions asked on the QA system have an answer within users' reviews, our goal is to determine such answer. Due to the extremely noisy setting we are operating in, we simplify the task of automatic question answering by limiting the scope of our project to binary (yes / no) questions only. The task of generating an answer becomes, in this way, a binary classification problem. This is a strong simplification but still allows our system to cover the majority of the inquiries. We utilize several unsupervised methods in order to extract the most relevant information from the reviews. We then train a neural model in a related task that shares structural similarities with the QA task and utilize the features produced by such a model to train our final classifier.

1 Introduction

During the early stages of online shopping, customers hesitated in trusting an electronic system because it lacked the human interaction of physical stores. The introduction of customer reviews brought a sense of community into the experience and played a central role in building this trust. Reviews create an open discussion that allows customers to share opinions and build a more personal and complete portrait of products. In fact, shoppers strongly rely on reviews when evaluating an item. For the purpose of our project, we focus our attention on one of the largest and most popular online shopping platforms: Amazon. Today, items sold on Amazon quickly accumulate a number of reviews that is too large to be tackled by a single reader and, when dubious about some aspects of the product, customers turn to the QA section. In this context, the task of automatically answering these questions based on the available reviews is of particular interest. To achieve our goal of answering binary questions on the Amazon QA dataset, we split the task into two separate problems: (1) Extraction of query-related sentences form reliable reviews. (2) Measuring the agreement between the sentences we have extracted and the query in consideration in order to infer a binary query response: "*yes*" or "*no*". When dealing with the Amazon review corpus, the quality of the data is a major obstacle, for the following reasons: (1) Even if the queries are labeled (we know the correct answer) there is no way to connect this answer with a particular review or even to guarantee that it can be found within the reviews. This limits our search to be performed with unsupervised approaches. (2) There are very few linguistic characteristics consistent across different reviews. (3) User-generated reviews contain a lot of noise, introduced from typos and bad grammar. (4) The reviews are highly subjective and, consequently, we will often find contradicting opinions regarding a specific query we want to answer.
Preprocessing of the reviews, identification of reliable reviews and extraction of relevant sentences are essential to reducing the amount of noise in our model's input.

1.1 Contributions

Our work focuses on building a pipeline capable of extracting relevant information in an unsupervised fashion. We employ several methods in order to retrieve relevant sentences related to product inquiries in a very noisy setting such as the Amazon Reviews dataset. We propose a new approach of encoding question-reviews pairs based on a neural feature extractor trained on a related task. Finally, we build a classification model for the binary Q&A task that uses as input the obtained representation of question and reviews, achieving interesting results.

2 Related Work

Automatic Summarization (Allahyari et al., 2017): this task shares with our work the goal of extracting relevant information within a document. Of particular interest within this field is the task of Multi-document summarization (Goldstein et al., 2000). Multi-document summarization consists of representing a collection of documents in a compact way capturing the relevant information while discarding non-relevant details. Most of the work in this field has focused on factual texts but there are interesting approaches that focus on summarizing product reviews (Zhang et al., 2012 & Di Fabbrizio et al., 2011) which are inherently biased and opinionated.

Information Retrieval and Relevance Ranking: Most of the work in the field of Automatic summarization is not query based though there are some exceptions, for example, Liu et al., 2017. Our approach uses a simple ranking method based on the upvotes/downvotes ratio of each review associated with Okapi BM25 (Jones et al., 2000), a state-of-the-art relevance ranking measure based on word level similarity.

QA Systems A lot of work has been done in this context, especially since task-specific large-scale corpora such as Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2018) and NarrativeQA (Kocisky et al., 2018) have been made available. Current state of the art on this tasks achieve almost human performance (Wang et al., 2018 & Seo et al., 2016). This setting differs from ours under two aspects: (1) the answer to a question is either available in the text or explicitly marked as unanswerable. (2) questions are about factual information not opinions and there are no contradicting answers in the text.

Opinion Mining Some of the most similar work to our own focuses mainly on studying ambiguity and subjectivity in customer reviews, for example McAuley and Yang, 2016 and Wan and McAuley, 2016. Their goal is to automatically learn whether a review of a product is relevant to a given query. The above work's main focus is to model ambiguity and subjectiveness in the answers and, while we concentrate more on the task of answering factual information about the products, we cannot prescind from this factors since they are characteristic of the domain.

3 The Datasets

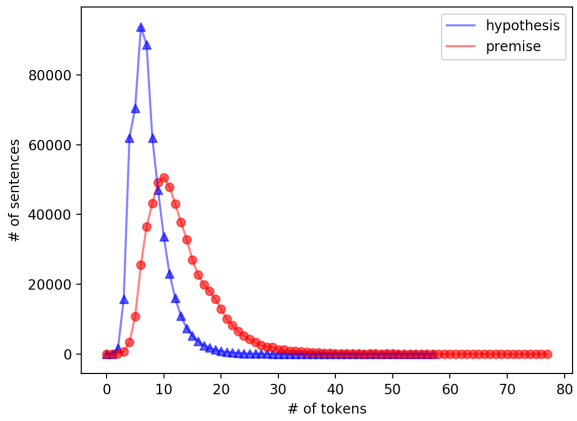
3.1 Stanford Natural Language Inference Corpus

The SNLI corpus (Bowman et al., 2015) is a collection of 570,000 English sentence pairs, in which each pair is made up of a "*premise*" and an "*hypothesis*". The *premise* is a short sentence describing a scene and the *hypothesis* is a sentence that can contain the same or part of the information of the premise or can be unrelated to it.
Each pair is labeled in one of the following three ways:

- *entailment*: the premise *entails* the hypothesis.
- *neutral*: the hypothesis is *neutral* with respect to the premise.
- *contradiction*: the hypothesis is in *contradiction* with the premise.

The premise sentences were obtained from preexisting corpora, while the hypothesis sentences and annotations were crowd-sourced on Amazon Mechanical Turk.
The generation and annotation phase was followed by a validation phase where 5 distinct annotators assigned a label to each sentence pair. The final label was determined through majority voting.
The dataset is balanced as in there are around 180,000 sentences pairs for each class.
The data provided by the Stanford NLP Group is already split into training, development, and test consisting respectively of 550000, 10000 and 10000 sentence pairs.
The average sentence length for the *premise* is 14.1 while for the hypothesis it is 8.3.
Figure (1) shows the distribution over the length of sequences.

Figure 1: Length distribution of sentences in the SNLI corpus.



3.2 Quora Dataset

The Quora dataset (Quora, 2016) is a collection of over 400,000 question pairs that are potential duplicates. The questions have been collected from the Quora website, a social platform for users to share knowledge and experience. Each line of the dataset contains an *ID* that is unique to the line, *id1* and *id2* that uniquely identify the two questions, the full text of the questions and finally a Boolean value "*is_duplicate*" that denotes whether the two questions are asking the same thing. The Boolean labels were assigned by field experts, nevertheless, they are inherently subjective due to the nature of the task and inevitably introduce some noise in the dataset. For the purpose of our project, we did not need the identifiers and therefore removed them. In table(2) we report an extract of the dataset.

3.3 Amazon QA and Reviews Dataset

The following two datasets (Wan and McAuley, 2016 & McAuley and Yang, 2016) contain information related to the same collection of products defined unambiguously by their identifier.
The Amazon question/answer dataset is a collection of roughly 1.4 million answered questions regarding amazon products. Each question can be a *yes / no* or an *open* question and the associated answer type can be a *yes*, *no* or *undefined*(for open-ended questions). For the purpose of our project, we only keep the binary (*yes / no*) questions.

Each entry is composed by the following values:

- Product ID
- Question
- Answer (y/n)

The reviews dataset contains approximately 18 million reviews of Amazon's products. Among all the possible values, we decided to keep the following: Each entry, we kept the following values:

- Product ID
- Review
- Review Summary
- Product Evaluation (1-5)

Table(4) shows an extract of the two datasets.

4 Methods

4.1 Natural Language Inference Model

The first step of our pipeline is training a model on the SNLI Corpus to classify sentence pairs as being in a relation of *entailment*, of *contradiction* or of *neutrality*.

Table 1: SNLI Corpus

Premise	**Hypothesis**	**Judgment**
An older and younger men smiling	The man is sleeping	Neutral
A soccer game with multiple males playing	Some men are playing a sport	Entailment
A black race car starts up in front of a crowd of people	A man is driving down a lonely road	Contradiction
A man inspects the uniform of a figure in some East Asian country	The man is sleeping	Contradiction

Table 2: Quora Dataset

Question 1	**Question 2**	**Duplicate**
How do I read and find my YouTube comments?	How can I see all my YouTube comments?	True
Why do rockets look white?	Why are rockets and boosters painted white?	True
What is a web application?	What is a web application framework?	False
What's one thing you would like to do better?	What's one thing you do despite knowing better?	False

Table 4: Amazon QA Dataset

ASIN	Query	Answer
B067EH7	Does this work on kindle fire?	Yes
B0N4SE8	Does this headset work with Linux?	No

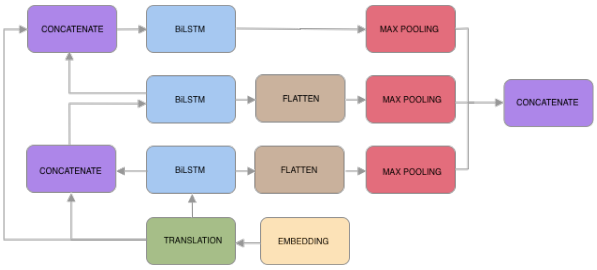
Preprocessing of the SNLI Corpus The inference Model takes as input vectors of fixed length. In order to generate sentence vectors we first created a dictionary containing all the words of the SNLI corpus and of the Amazon Corpora. For each sentence of the corpus we first performed tokenization and filtering of punctuation, then we generated a vector of size 40, where each element holds the dictionary index of the word present in the sentence at the corresponding location. Sentences longer than 40 words were removed entirely from the dataset since they only accounted for 2% of the total number of instances. Vectors representing sentences shorter than 40 words were left-zero-padded. We also removed samples for which there was not a consensus among the annotators on the ground truth, accounting for another 2% of the available data.

Architecture For this task of Natural Language Inference we decided to use a neural model as shown in figure 3. The architecture follows a sentence-embedding approach as described in Bowman et al., 2015, figure 2 shows a more detailed representation of the structure of the encoder.

At first, each token in the input sequence is converted into its word embedding representation, we used GloVe (Pennington et al., 2014) Embeddings with 300 dimensions. Then, like in Bowman et al., 2015 we have a "Translation" layer which allows learning a modified version of the input embeddings without updating the weights of the embeddings themselves. Following the same reasoning of Sun et al., 2017 we decided not to train the embedding weights so as to increase the generalization capabilities of the model on unseen data.

The following part of the encoder is similar to the hierarchical structure described by Conneau et al., 2017, we use a stack of 3 Bidirectional LSTM (BiLSTM) (Schuster and Paliwal, 1997) layers with a concatenation of the output of the previous layer and the original input sequence as input. In order to mitigate overfitting, we apply recurrent dropout (Gal and Ghahramani, 2016) to the BiLSTM layers. We then apply a max pooling operation over the output sequence (along the time dimension) of each BiLSTM layer, the concatenation of the resulting vectors represents the sentence embedding. The encoder part of the network is duplicated and used to produce the sentence embeddings for both the premise and the hypothesis.

Figure 2: Encoder Architecture



These two embeddings are combined in order to produce simple similarity measures such as element-wise difference and element-wise product as described in Mou et al., 2016. Finally, we have a fully connected (FC) portion ending with a 3-way Softmax output that defines a distribution over the target variable. As activation function for the neurons of the FC layers, we used Leaky ReLU (Maas et al., 2013) in order to preserve gradient flow.

Goal Our main objective in developing this component is to build a model capable of creating a valid representation of the agreement between two input sentences, which will be used as a feature extractor for the Q&A classification model.

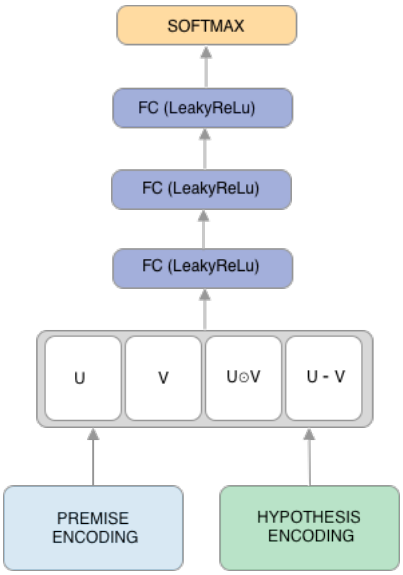
4.2 Validation on Quora

Using the same preprocessing and the same architecture we used for the NLI task, we tested the performances of the model on this different domain, both using the previously trained model as a fea-

Table 3: Amazon Review Dataset

ASIN	Review	Helpful
B06IJPQ	I had 3 of these disks on a Windows 7 System setup in a Dynamic Spanned configuration before I started to have problems	35/57
B144X3G	They seriously have every size. They are easy to store, only thing I didn't like about them is that the ring that keeps them together	0/0

Figure 3: NLI Classifier Architecture



ture extractor and also performing fine-tuning.

Goal Our objective here was not to achieve state-of-the-art performances on this task but to verify whether or not the encoding representation produced by the NLI model was applicable to other domains, the results reported in 5.2 suggest that this is the case.

4.3 Review Ranking and Sentence Extraction

Our approach aims to use the Inference Model we just described to compare each query we want to answer, with a number of selected sentences extracted from the reviews of the same product the query refers to. The sentences we extract should be as closely related as possible to the query and, ideally, should be extracted from objective and reliable reviews. The first step we take is therefore to rank the reviews in an effort to prune "bad" reviews and reduce the number of contradicting sentences we extract, and improving the reliability of our comparisons. The review ranking is based on the *Wilson Score Interval* (Wilson, 1927):

$$Confidence(p, n, z) =$$

$$\frac{p + \frac{z^2}{2n}}{1 + \frac{z^2}{n}} \pm \frac{z}{1 + \frac{z^2}{n}} \sqrt{\frac{p(1-p)}{n} + \frac{z^2}{4n^2}}$$

where p is the observed fraction of positive ratings, n is the total number of ratings and z determines the target confidence. This metric estimates the "confidence" of how relevant a review is, based on the number of upvotes/downvotes it has received. It is, for example, used for comment ranking by Reddit. This estimation allows us to surface the most reliable reviews. We set a threshold which leaves us with 50 to 100 reviews per product.

The second step is the extraction, from the selected reviews, of the sentences that have the strongest correlation with the query. This is done to reduce the amount of noise that is fed to the SVM model. Since a query is only interested about a specific aspect of the product, only a small fraction (if any) of the sentences we find in the reviews will be of interest; the rest is just noise, which we want to remove as much of as possible. Nevertheless, the window of acceptance should be large enough to allow the sentences we actually want to include to be selected by the (imperfect) extraction process. To achieve this, we firstly split the reviews in a list of sentences using a *Sentence Tokenization* algorithm (Kiss and Strunk, 2006) available on the NLTK (Bird et al., 2009) platform. The *NLTK Sentence Tokenizer* uses an unsupervised approach that aims at correctly distinguishing between sentence boundaries and abbreviations. It is context independent and therefore fits well for our case.

Once the reviews have been split into sentences, we rank these using the *Okapi BM25* (Robertson and Zaragoza, 2009) metric:

$$score(D, Q) =$$

$$\sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k+1)}{f(q_i, D) + k \cdot \left(1 - b + b \cdot \frac{|D|}{Avg_{dl}}\right)}$$

where D is a document (a sentence in our case) and Q is the query (a question). BM25 is a ranking function that gives a reward based on the times a word is present in a sentence (*Term Frequency*, $f(q_i, D)$) but also leverage this quantity based on the rarity of the word among documents (*Inverse Document Frequency*, $IDF(q_i)$). Moreover, this metric introduce a penalization term for long documents ($\frac{|D|}{Avg_{dl}}$) where Avg_{dl} is the average lenght of all the documents. Finally, the terms b and k are parameters which we set as described in (Manning et al., 2008).

To optimize the performance of the ranking procedure, we preprocessed the query and the reviews:

- **Conversion to lowercase**

- **Removal of stop words and punctuation:** comparison of each word with the NLTK stop word collection for the English language.
- **Stemming:** NLTK Porter Stemmer (Porter, 1997)

This preprocessing stage improves BM25 matching which is based on word similarity. For example, if we have two sentences:
"Can I use this for walking my dog?" and "I walk my dog with it", only "dog" is a matching word. After cleaning and stemming, both "dog" and "walk" will match. We select from the extracted sentences the 10 sentences with the highest BM25 score. Each of the 10 selected sentences is transformed into its vector representation ("*tokenization*") and appended to the query vector. These tokens are given as input to our Inference Model that will generate their respective embeddings.

We considered different approaches to retrieve relevant information from the reviews: TextRank by SummaNLP (Barrios et al., 2016), POS-tagging based approaches and pruning highly subjective sentences (Wiebe and Riloff, 2005). BM25 ranking was chosen over these approaches due to the high degree of generalization we need for the task.

4.4 Query Binary Classifier

The final component of our pipeline is a classifier model that takes as input the features extracted from the Inference Model and predicts the binary output (*yes / no*). For this task, we used the implementation of Support Vector Machine (SVM) Classifiers available on Scikit-Learn (Pedregosa et al., 2011).

The extraction of the features generated by the Inference model happens just before the fully connected layer. What we find here is an encoding which contains the features extracted from one of the 10 query-sentence tokens generated in the Sentence Extraction phase. Ten instances of the Inference Model are run in parallel for each of the 10 query-sentence pairs generated, so what we end up with are 10 separate encodings. Each of these encodings contains the information of the query and one of the ten sentences extracted from the reviews. These encodings are compressed into a single vector by averaging over the values and the resulting single encoding is given as input to the SVM. The model is trained using the binary label we have for the given query.

We used 5-fold cross-validation to evaluate the performance of the model and to choose the best hyperparameters. The best configuration we found is the following:

Kernel	Radial Basis Function (rbf)
Gamma	1/20400
Error Penalty	1.24

5 Results

In this section, we present the result obtained by the different models described in the previous sections.

5.1 Metrics

		Predicted class P	**Predicted class** N
Real class	**P′**	True Positive (TP)	False Negative (FN)
	N′	False Positive (FP)	True Negative (FN)

We evaluate all the methods in term of *accuracy score*, i.e. the ratio between the correctly classified examples with respect to all the samples,

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Furthermore, for the binary classification model we evaluate the performance based on the *F1-score*

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

where, the *Recall* is the ratio of correctly predicted positive observations to the all observations in actual class, and the *Precision* is the ratio of correctly predicted positive observations to the total predicted positive observations, formally

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Finally, we also used the *Area Under The Curve (AUC)* score, that compares the True Positive Rate

(TPR) (or sensitivity), that is equal to the Recall, with the False Positive Rate (FPR) (1-specificity)

$$FPR = \frac{FP}{FP + TN}$$

5.2 Inference Model Results

In this section, we present the results obtained by the Natural Language Inference Model described in section (4.1).
We tested the model on the test section of the dataset, composed of 10000 examples.
We compare our model against state-of-the-art methods described in (Bowman et al., 2015) and (Kim et al., 2018).

Table 5: Results on the SNLI dataset

	Accuracy
Random Baseline	0.33
LSTM (Bowman et al., 2015)	0.77
Ensamble (Kim et al., 2018)	0.90
Our Method	0.81

As described in section (4.2), we also evaluate the performance of the inference model on the Quora Dataset. The accuracy we obtained with this model met our expectations.
We present the results obtained using the inference model with fixed encoder and with fine-tuning on the Quora Dataset, on a test set of 40000 samples. Since we are only interested in evaluating the performance of the model in extracting relevant features, we only compare the results of the model against the random and majority class classifiers.

Table 6: Results on the Quora dataset

	Accuracy
Random Baseline	0.50
Majority Class	0.66
Fixed Encoder	0.75
Fine Tuning	0.80

5.3 Binary Classifier Results

We evaluate our model taking into consideration its ability to correctly classify binary questions.
In particular, we show the results obtained with different configurations of the method, which are

- **Best reviews**: with this configuration, we only take into consideration the best reviews obtained using the method described in section 4.3, without splitting the reviews into sentences.
- **Best reviews with split on sentences**: in this case, we use the best reviews and we split them using the *Sentence Tokenizer*.
- **Final Model**: this model takes into consideration the best reviews and also uses the best phrases extracted using the *BM25* algorithm.

The results shown in table 7 are obtained using 5-fold cross-validation on the balanced dataset composed by 700 examples for each class (*yes/no*). Moreover, we applied paired t-test for statistical significance and we obtained a *p-value* score of 0.1.

6 Conclusions and Future Work

In the course of this work, we developed an unsupervised pipeline capable of extracting relevant information from a very noisy setting such as Amazon reviews. We tested the efficacy of neural models to learn representations of the input that can be applied to other domains. We then used such representation of question-reviews pairs to train a final classification model.

Our results are not strictly comparable with other examples in the literature due to some differences in the data preprocessing. Nevertheless, such results are still promising and the strategies we employed merit further study.

6.1 Future Work

There are several components of our system that can be further improved to achieve better overall performances. The accuracy of the NLI model can be increased as demonstrated by the several results available on the Stanford NLP Group website. Improving such model would allow the extraction of more relevant features from question-reviews pairs. The sentence extraction task employs an unsupervised algorithm for sentence boundaries detection. In this work, we used the pre-trained model available on NLTK so fine-tuning such model on the Amazon Reviews dataset could yield better results. Finally, there are several advantages to using unsupervised methods but it makes evaluating such approaches complex. Furthermore, since the models were run on unlabeled data, we had to speculate that the questions taken as input,

Table 7: Results on the Amazon Q&A dataset

	Accuracy		**F1-score**		**AUC**	
	μ	σ	μ	σ	μ	σ
Random Baseline	0.50	-	0.5	-	0.5	-
Best reviews	0.58	0.02	0.57	0.02	0.62	0.01
Best reviews with phrase split	0.60	0.03	0.58	0.04	0.63	0.04
Final Method	0.61	0.02	0.59	0.02	0.65	0.01

can, in fact, be answered with information found in the reviews. The results we obtain are tied to this assumption. Annotating at least part of the available data would allow for objective quality measures and more reliable results.

7 Appendix

Guglielmo Menchetti worked on the Amazon Corpus. He developed methods to efficiently manage the large amount of available data using Amazon EC2 instances to store the data in a relational DBMS. He wrote the programs that performed data cleaning and data transformation for all the experiments we carried out. He also contributed to the SVM classifier.

Lorenzo Norcini focused on the design and implementation of the neural models. He used Keras Framework and Google Colab platform to build and train the models. He also performed data cleaning and preprocessing on the SNLI and Quora corpora. He also contributed to the SVM classifier.

Federico Sandrelli Researched and tested several methods of sentence extraction and ranking, wrote the program to perform Okapi BM25. He also contributed to the SVM classifier.

Team We all worked together in order to define the problem and how to better tackle it. We performed independent researches and then discussed our findings. Each of us contributed to this paper describing the part of work they focused on.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth Trippe, Juan Gutierrez, and Krys Kochut. 2017. Text summarization techniques: A brief survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 8:397–405.

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics.

Giuseppe Di Fabbrizio, Ahmet Aker, and Robert Gaizauskas. 2011. Starlet: Multi-document summarization of service and product reviews with balanced rating distributions. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 67–74, Washington, DC, USA. IEEE Computer Society.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 40–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

K. Sparck Jones, S. Walker, and S. E. Robertson. 2000. A probabilistic model of information retrieval: Development and comparative experiments. *Inf. Process. Manage.*, 36(6):779–808.

Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak. 2018. Semantic sentence matching with densely-connected recurrent and co-attentive information. *CoRR*, abs/1805.11360.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Tom Kocisky, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, TBD:TBD.

Mengwen Liu, Yi Fang, Alexander G. Choulos, Dae Hoon Park, and Xiaohua Hu. 2017. Product review summarization through question retrieval and diversification. *Inf. Retr.*, 20(6):575–605.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *WWW*, pages 625–635. ACM.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

M. F. Porter. 1997. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Quora. 2016. Quora question pairs.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Chengjie Sun, Yang Liu, Chang'e Jia, Bingquan Liu, and Lei Lin. 2017. Recognizing text entailment via bidirectional LSTM model with inner-attention. In *Intelligent Computing Methodologies - 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, Proceedings, Part III*, pages 448–457.

Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *ICDM*, pages 489–498. IEEE.

Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714. Association for Computational Linguistics.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *In CICLing2005*, pages 486–497.

Edwin B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

D. Zhang, J. Ma, X. Niu, S. Gao, and L. Song. 2012. Multi-document summarization of product reviews. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1309–1314.