Text Analytics Group Project - NLP-Based Twitter Chatbot for Customer Support

ISGB/BYGB7977 Spring 2021: Text Analytics
Prof. Yilu Zhou

Group 8:
Akshit Arora
Prannoiy Chandran
Chandni Khattar
Hao Lyu
Xuechun Jiang

Executive Summary

The project used data extracted from 3 millions of tweets and replies from companies to build a customer service chatbot to automatically reply users' queries, helping companies deal with customer concerns and improving customer satisfaction. The goal was to find requests from customers similar to the one provided as a query and return answers using Natural Language Processing (NLP) techniques.

There are two major steps in building the chatbot using a machine learning based approach. The first step was text preprocessing: we cleaned tweet data, removed symbols, and joined the questions and answers into the same row to train the model. We used the tweets posted from and to the three most responsive companies Amazon, Apple and Uber only. The next step was feature extraction and modeling: we designed three classes (Seg, Sentence and SentenceSimilarity) and four algorithms (TFIDF, LSI, LDA, and Word2vec) to build models. The chatbot returns up to five most relevant answers from a library of predefined responses, based on the similarity scores between the input question and similar questions answered by the customer service teams.

Apart from the chatbot, we also applied topic modeling. To better learn customers' needs, we used LDA to extract 10 topics from customers' tweets to Amazon. These topics can also reflect users' major concerns. Therefore, we used the most popular terms to generate input questions to evaluate the performance of the four models.

Similarity scores and runtime were used as the primary performance metrics, along with qualitative analysis of the answers returned. LSI was found to have the best performance in providing the most relevant answer within the shortest time. Potential improvements for future deployments were also discussed.

Problem statement

In today's world, customer experience is an integral part of business life. Since some companies have realized the advantages of responding to comments on social channels, namely Instagram and Twitter, they are meticulously monitoring the comments in order to increase customer satisfaction. According to Hyken (2016), companies using Twitter as a social channel are able to get a 19% increase in their customer satisfaction.

Customer support chatbots are getting increasingly sophisticated due to widespread adoption. Since manual data analysis requires a huge amount of time and excessive resources, building a chatbot offers numerous benefits. A chatbot provides convenient service to many companies who desire to respond to customers' queries immediately and properly, aiming to reach maximum customer satisfaction and improve their user experience. For this purpose, customer support chatbot firstly predicts the sort of questions that a user may ask or the manner that will be possibly raised and then it gives an answer properly according to customer queries. Companies can also extract meaningful information from text to identify customer pain points.

Data Description

In this study, the dataset downloaded from Kaggle includes 3 million tweets and replies from the biggest brands (e.g. Apple, Amazon, Uber) on Twitter. Every conversation has at least one request from a consumer and at least one response from a company.

Attribute	Explanation
tweet_id	A unique, anonymized ID for the Tweet
author_id	A unique, anonymized user ID
inbound	Whether the tweet is "inbound" to a company doing customer support on Twitter
created_at	Date and time when the tweet was sent
text	Tweet content from customers and companies
response_tweet_id	IDs of tweets that are responses to this tweet
in_response_to_tweet_id	ID of the tweet this tweet is in response to

Figure 1. Variables and Explanation

The dataset consists of seven columns, which are tweet_id, author_id, inbound, created_at, text, response_tweet_id, and in_response_to_tweet_id.

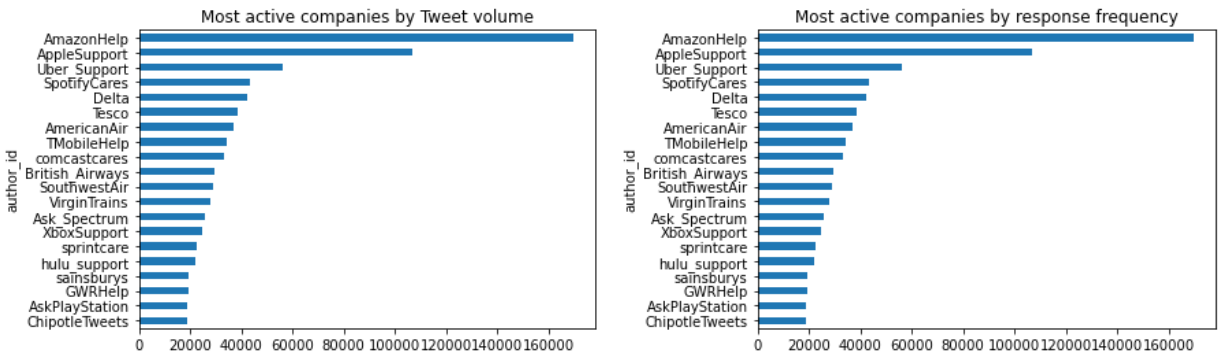

Figure 2. Companies ranked by customer service activity on Twitter

The most active companies by both Tweet volume and response frequency are largely the same, with Amazon, Apple and Uber ranking as the 3 most active companies. Amazon and Apple each responded to over a 100,000 Tweets. This underscores the modern prevalence of ecommerce, personal devices and consumer applications, as well as the need for customer service departments to be prepared to handle a large volume of customer complaints over social media channels. With the increasing adoption of mobile devices and social media, it is likely that phone-based customer service will decline in importance while customer engagement over a variety of online channels will form the bulk of customer support.

Business Goal Analysis

Twitter is becoming an increasingly popular and prominent tool for companies to engage with customers and for customers to raise concerns. Many customers use Twitter nowadays to engage with customer service departments due to its convenience and ease of use. Alternative ways to engage with customer service agents, such as making phone calls or sending emails, are not always ideal. Being on hold for extended periods of time is common when making a phone call and with emails, it may take 2-3 emails for the problem to be explained and resolved. Twitter may help to reduce the time lag that is inherent in other customer service channels.

The quality of customer service provided is a crucial aspect of any consumer-facing business. Companies are often judged on metrics such as their churn rates or net promoter scores, and stories of negative experiences with a company can easily go viral. In almost all cases, customers only engage with customer service channels when they have an issue or a question, not when they are satisfied with the service they have received or they are clear about what they have to do. Therefore, ensuring a positive experience is essential. However, companies are limited in how many customer service agents they can hire, and these agents may have to handle multiple customers at a time. There is an increasing need to make the customer service process on Twitter faster and more efficient using automation.

The business goal is to build a system that will reduce time and costs for both companies and customers. To achieve this goal, a chatbot will be built to handle most types of customer messages. Chatbots are programs that can automatically engage with messages from customers. They can respond differently to messages containing different topics and key words and quickly offer solutions or navigate customers to human agents or websites for further help. The functional requirements for the chatbot are as follows: (1) acknowledge the customer's issue or query, (2) provide potential answers and if needed, redirect the customer to other resources, (3) provide responses to similar questions.

Methodology

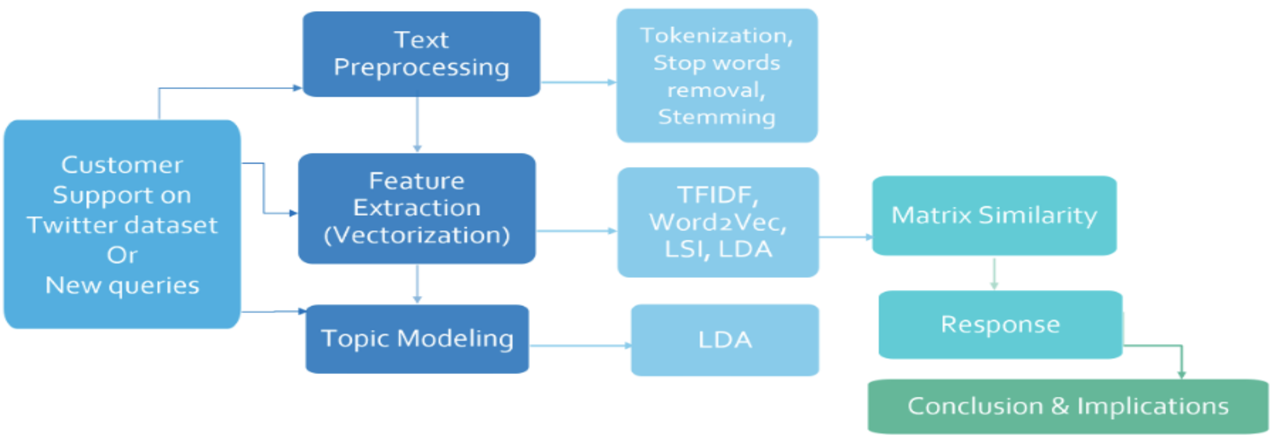

Figure 3. Methodology Flow Chart

The dataset was pre-processed (tokenization, stopwords removal and stemming) before topics were modeled and features were extracted. Principal component analysis was performed on the output from topic modeling. After the chatbot's responses were observed in multiple test sessions, conclusions were derived on the system's usability and scope, as well as potential improvements.

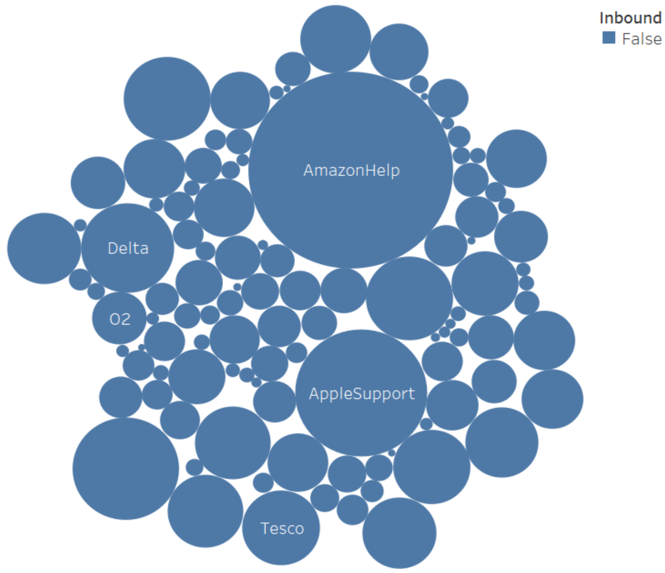

Figure 4. Bubble Chart of company activity on Twitter

In the visualization of how active companies are in responding to Tweets, inbound set to False means that the Tweet content is not sent to a company, which in effect means the account sending the tweet is a corporate customer service account. From this bubble chart, Amazon's Help account exceeded all other companies in the number of customer support Tweets, which corroborates the observations from the bar charts. Apple Support, Delta, and Tesco are also active customer support accounts on Twitter as shown. The insight behind is that the larger the bubble, the more people need these customer service channels, which further implies the importance of developing an efficient chatbot for customer service for these companies.

The observations thus far suggest that chatbots will frequently answer questions related to e-commerce (delivery time, package tracking, cancelling or modifying orders, refunds, etc.) and personal devices (repairs, warranty, technical support, etc.). Therefore, these are topics that the chatbot's performance must be tested in.

Figure 4 shows how active customers are requesting customer service on Twitter. Most of the customers have less than 50 conversations with customer service accounts within the dataset range. From this chart it can be seen that there are customers who are asking for help for as many as 454 requests and replies which is astonishing. The expenses of communication for companies could be unsustainable if there is a large volume of Tweets to deal with, which leads to the goal of the project, to build a chatbot that would not only reduce costs, but also offer customers with precise answers. Most of these interactions can be automated, leaving human agents free to handle complex cases and reducing the time they would spend gathering contextual information.

System Implementation

a. Topic Models

Topic modeling is the process of identifying topics in a set of documents, which is very useful in supporting customer service automation where knowing the topics is important.

The Latent Dirichlet Algorithm (LDA) was imported from the Gensim library and implemented using Python to identify topics in Tweet contents from the dataset. LDA is a form of unsupervised learning which generally views documents as bag-of-words, breaking up the document into every single word. LDA makes the assumption that the document is generated by picking a set of topics, with topics picking a set of words.

Before applying the tweet contents to the LDA topic model, data cleaning was performed, including replacing special characters like '/r', leaving only English characters without punctuations, converting text into lowercase, removing stop words like 'is', 'are', and tokenizing the text which also created a list of words. Words that occurred only once were also removed. To generate a term document matrix, a token dictionary class and a corpus were created.

	0	1	2	3	4	5	6	7	8	9
0	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000
1	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000
2	0.016667	0.016667	0.849990	0.016667	0.016667	0.016668	0.016668	0.016668	0.016668	0.016667
3	0.025004	0.025002	0.352088	0.025002	0.447889	0.025002	0.025003	0.025004	0.025003	0.025002
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.731660	0.000000	0.206780	0.000000
...	...	...	...	...	...	...	...	...	...	...
84561	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000
84562	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000
84563	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000
84564	0.021662	0.021662	0.021662	0.021662	0.021662	0.021662	0.021662	0.805039	0.021662	0.021662
84565	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000	0.100000

84566 rows × 10 columns

Figure 5: Term document matrix (Amazon)

In the sample matrix (using Amazon as a case study), there are 84,566 records in total related to Amazon. For each of the tweet content, the dominant topic ranges from 0 to 9 derived from Figure 5 was assigned a score based on relevance with each topic. The distribution of each topic is meaningful in a way that it tells which topic is dominant over the document, or the tweet content in this situation. Each row of the result can be referred to as per-document topic distribution. A symmetric distribution would mean that each topic is evenly distributed in the tweet content, while asymmetric distribution tells that certain topics are dominant in the tweet content.

b. Natural Language Processing (NLP)-based Chatbot

As is common with NLP tasks, the chatbot pipeline was initiated by converting words into tokens for handling as n-grams and documents (text objects; in this case, sentences and phrases from Tweet contents). Further preprocessing was performed to remove stop words and stem tokens using the Porter stemmer.

Cosine similarity was used to compute similarities between words and sentences. This technique uses the relative orientation of 2 non-zero vectors to determine how close or similar they are. This accounted for the large size of the dataset. Even if 2 documents were far apart in terms of Euclidean distance (such as by being separated by thousands, or even hundreds of thousands of rows in the dataframe), the orientation angle may have been close enough to reveal a degree of similarity between them.

Four encoding models were imported from Gensim and implemented using Python for the user to choose from: (1), Word2vec, (2) Latent Semantic Indexing (LSI), (3) LDA and (4) Term Frequency-Inverse Document Frequency (TF-IDF).

Word2vec uses a neural network to learn word associations, representing each word as a vector. Semantic similarity between vectors is then computed using cosine similarity.

LSI, which uses singular value decomposition (SVD) to identify patterns in unstructured text, is based on the principle that words used in similar contexts often have similar meanings.

The TF-IDF value increases proportionally to the number of times a word appears in a document, but also uses inverse document frequency to offset the number of documents in a corpus that contain the word. This identifies words that occur in few documents, but are frequently mentioned in the documents they do appear in.

The final chatbot was implemented using Tweets collected from the 3 most responsive companies on Twitter: Amazon, Apple and Uber.

Evaluation

a. Topic Models

```
[(0,
  '0.097*"service" + 0.086*"customer" + 0.017*"worst" + 0.015*"disappointed" + 0.015*"support" + 0.014*"bad" + 0.012*"experienc
e" + 0.011*"time" + 0.011*"even" + 0.011*"really"'),
 (1,
  '0.021*"done" + 0.017*"fire" + 0.014*"alexa" + 0.014*"mail" + 0.012*"front" + 0.011*"tv" + 0.011*"mal" + 0.011*"thought" + 0.
010*"sold" + 0.010*"anyone"'),
 (2,
  '0.037*"please" + 0.035*"order" + 0.026*"help" + 0.021*"prime" + 0.014*"account" + 0.014*"need" + 0.014*"india" + 0.012*"canc
el" + 0.012*"get" + 0.011*"available"'),
 (3,
  '0.055*"order" + 0.034*"received" + 0.026*"still" + 0.022*"yet" + 0.019*"refund" + 0.017*"product" + 0.016*"ordered" + 0.014
*"delivery" + 0.013*"date" + 0.012*"havent"'),
 (4,
  '0.042*"delivered" + 0.027*"today" + 0.026*"package" + 0.024*"thanks" + 0.023*"order" + 0.016*"delivery" + 0.016*"says" + 0.0
13*"yesterday" + 0.011*"said" + 0.011*"tracking"'),
 (5,
  '0.041*"email" + 0.021*"call" + 0.018*"contact" + 0.015*"times" + 0.014*"details" + 0.013*"sent" + 0.013*"link" + 0.013*"alre
ady" + 0.012*"account" + 0.010*"phone"'),
 (6,
  '0.020*"back" + 0.016*"return" + 0.015*"card" + 0.015*"refund" + 0.015*"money" + 0.014*"get" + 0.013*"order" + 0.013*"kindle"
+ 0.013*"got" + 0.012*"ordered"'),
 (7,
  '0.022*"gracias" + 0.019*"merci" + 0.015*"logistics" + 0.013*"colis" + 0.011*"pedido" + 0.011*"prime" + 0.010*"cest" + 0.010
*"ok" + 0.009*"bien" + 0.009*"watch"'),
 (8,
  '0.052*"delivery" + 0.041*"prime" + 0.033*"day" + 0.023*"time" + 0.020*"package" + 0.020*"shipping" + 0.019*"days" + 0.015*"g
et" + 0.012*"two" + 0.012*"today"'),
 (9,
  '0.047*"thank" + 0.026*"thanks" + 0.018*"replacement" + 0.016*"much" + 0.015*"working" + 0.015*"usps" + 0.014*"problem" + 0.0
14*"plus" + 0.013*"app" + 0.013*"happy"')]
```

Figure 6. Sample LDA topic distribution (Amazon)

The sample topic distribution used Amazon Help's Tweets as the input for the LDA model . The number of keywords was set to 10 in the code, so the output displays the top 10 keywords of dominant topics, with each keyword being assigned with a weight score. Each row of the result can be referred to as a per-topic word distribution. Here a certain topic is referred to according to its keywords. Taking topic 0 as an example, users talk about "service", "customer", "worst", "disappointed"; the topic maps to "Bad Experiences with Customer Service".

Companies can study these topic models to gain insights on areas to improve. For example, topic 4 features words such as "package", "tracking" and "yesterday". These complaints are to be expected from customers of an e-commerce company. If Amazon frequently receives such complaints, it may have to audit its supply chain or delivery processes. Meanwhile, topic 6 frequently includes "card", "refund" and "return". If this topic ranks highly in marginal topic distribution (the relative importance of each topic), it would be a sign for the company to review its payment and refund procedures to reduce the number of payment-related complaints and queries.

However, interpretations must be carefully considered based on the context of customer service interactions. For instance, topic 7 includes the word "thanks" in 2 languages (French and

Spanish). While this seems to imply that customers were satisfied with the service they received, it may just be from the word being used commonly when interacting with someone online.

Similarly, the meaning of “happy” in topic 9 differs greatly depending on if it was used as a single word or if it was part of a bigram (“not happy”). Further refining of the model may be performed for future use cases to examine the context in which such words appear, primarily by using n-grams or capturing these words’ part-of-speech (POS) tag.

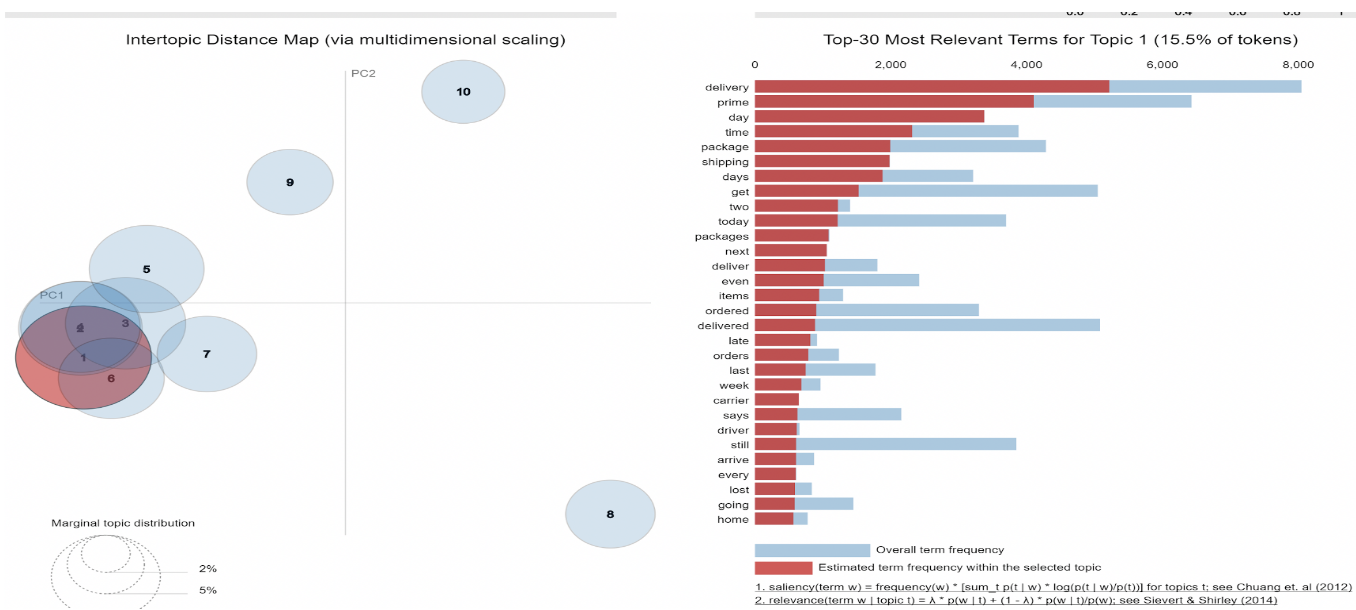

Figure 7. Interactive Visualization of Topic Modeling using LDAvis

The distance map (output from the pyLDAvis library) allows for principal component analysis and indicates that most topics had several terms in common. According to Chuang et al. (2012), the centers of the circles in the intertopic distance map are determined using multidimensional scaling to project the inter-topic distances onto two dimensions. The area of each circle represents the overall prevalence of each topic.

Topics 1 to 7 are closely related and share multiple common terms in the corpus because of the overlaps among these topics from the distance map. Topics 8 and 10 were notably not as similar to the others. The salient terms in topic 1 largely pertain to delivery, packages and shipping. With topic 1 closely clustered with several other topics on the map, it is implied that delivery and shipping-related concerns were heavily represented in other topics as well, and therefore across most complaints or queries received by Amazon Help.

b. NLP-based Chatbot

```
Hey there! This is Auto Customer Service. First, please choose the company you would like to chat with:
0 for Amazon, 1 for Apple, 2 for Uber, 'q' to quit: 0
Please type your question here or press 'q' to quit: can i cancel my prime membership
Thanks for asking, here is the 1 most likely answer(s) from AmazonHelp:
for the next year is added to your cart automatically.Please refer https://t.co/9OJHAgxsHU to learn more about the pr
ogram(2/2)^SF
Is your problem solved? (input y/n)n

Thanks for asking, here is the 2 most likely answer(s) from AmazonHelp:
As soon as your Prime Membership expires, email will be sent to you informing that the option to purchase Prime Membe
rship(1/2)^SF
Is your problem solved? (input y/n)n

Thanks for asking, here is the 3 most likely answer(s) from AmazonHelp:
That's a great question! You can learn more about how to switch membership plans here: https://t.co/5YKBDL4sXl ^HC
Is your problem solved? (input y/n)n

Thanks for asking, here is the 4 most likely answer(s) from AmazonHelp:
We're here to help! You can cancel Prime by clicking on this link: https://t.co/3hUQfcBm7s ^GL
Is your problem solved? (input y/n)y
```

Figure 8. Sample chatbot session 1 (Amazon, Word2vec)

In the first example of a chatbot session (which was run with Word2vec as the encoding model, and addressed towards Amazon Help), the user's question leads to a number of potential answers, with the fourth answer directly answering the user's question. The chatbot acknowledges the user's specific issue (in this case, cancelling their Amazon Prime membership), and provides links to pages where the user can resolve their issue. The resources that the chatbot directs users to may include links to frequently-asked-questions pages or links to contact the company by other means (via phone or direct message). A straightforward question such as this being answered by a chatbot, and the user being provided with clear instructions on how to independently resolve the issue, saves both the user and the company from a lengthier process over the phone.

According to the most relevant terms generated from Topic 1, the package delivery time is a major concern for most Amazon users. Therefore, one frequently-asked question for the customer service could be 'I selected free two day delivery but the package arrived late', which was used as a query to test the results of the chatbot.

Model	First Returned Answer	Most Similar Question	Similarity Score
Word2vec	Thanks for the update, apologies again for the delay. ^JJ	All items arrived by last night, one day past the "guaranteed" delivery date, but I have them	0.9572
TFIDF	Oh my, in this case, feel free to call or chat us when you have the time.	Its two days late tho	0.5529
LDA	I completely understand your frustration! We'd like to take a close look at the issue. Please reach out to us directly via phone/chat	It should've been delivered today by 3. So Amazon is going on 4 days late and UPS is going on 2 days late. I was supposed to get my package on Sat Nov 25.	0.8935
LSI	Terribly sorry about that! Could you please let us know when that expected delivery date was? If a package doesn't make the original delivery date, it usually arrives the day after, but if it has been later than that, we'd like to further assist. Let us know!	Frustrated. When I pay extra for 2 day shipping, I would really like the package to arrive on the expected delivery date, @115821. But no. Here I am, package-less.	0.6520

Figure 9. Chatbot responses from each encoding model (Amazon)

The table summarizes the first responses returned by the four models and the most similar questions based on the similarity scores, with the same input query used for each of the 4 models. Each model returned a different set of responses based on the different encoding methods for calculating similarity distances. Similarity score was used as a performance metric here. As each question returns a list of potential responses, it is also useful to compare the scores for potential responses within the same encoding method.

Word2Vec, LDA, and LSI models returned five responses with similarity scores above the threshold 0.5, whereas the TFIDF model was able to return only two relevant responses. Unlike the other three models, TFIDF did not answer the query well as the two most similar questions are not so similar to the input query. In fact, the third question scored 0.4 was more relevant than the first, but the answer was not returned. Therefore, the threshold should be set lower to around 0.4 so that the chatbot will answer the query better.

Word2Vec returned five similar questions with similarity scores all above 0.95. As seen in the sample chatbot session for Amazon, the five responses are relevant to package delay and can solve the input query, but the similar questions suggested by Word2vec were not exactly the same question. The threshold should be set higher but we need to test more queries to determine an ideal threshold. LDA also answered the input query well with the first four returned answers scored above 0.85. The last question scored below 0.85 was not similar to the input query. We may consider setting the threshold to around that level, but more testing is required for adjustment. The first answer returned by the LSI was the best among the four models in terms of relevance and professionalism. In fact, the most similar question returned by LSI delivered the same meaning as the input question, although the similarity score of 0.65 did not seem high.

In general, Word2vec and TFIDF are more applicable to analyze short text. LDA and LSI have better performance in dealing with longer text. Whereas LDA may take a longer time to run, LSI is the best algorithm for handling different lengths of text efficiently.

The model can also be evaluated based on the time it took to return answers. The set up time is not as relevant because setup only needs to happen once, but the runtime is particularly useful as a performance metric. It is understandable that these times may vary each time the model is run and for different queries, but the differences should be minimal. Since the cutoffs are under the same device condition, the measures in the below table are applicable to other scenarios.

Model	**Set Up Time (seconds)**	**Time to Return Results (seconds)**
Word2vec	257	0.982
TFIDF	297	0.320
LDA	522	0.075
LSI	299	0.082

Figure 10. Setup times and runtimes for each chatbot model (Amazon)

There seems to be an inverse relationship between the set up time and time for returning results. Word2vec has the shortest set up time but the longest running time, whereas LDA has a double set up time but less than 10% of the running time. Both TFIDF and LSI took about 300 seconds to set up, but LSI was much more efficient in returning a relevant and professional answer.

```
----------------------------------------------------------------------------------------------------
Hey there! This is Auto Customer Service. First please choose the company:
0 for Amazon, 1 for Apple, 2 for Uber: 1
Please type your question here ('q' to quit): Where can I repair my iPhone X?
Thanks for asking, here is the 1 most likely answer(s) from AppleSupport
We're happy to help. We can look up some service options. Let's work together in a DM. https://t.co/GDrqU22YpT
```

Figure 11. Sample chatbot session 2 (Apple, LDA)

In the second chatbot test session, LDA was used as the encoding model and questions were directed towards Apple Support. This session was used to test the chatbot's performance when faced with questions about hardware and software support. When asked about repairing an iPhone, the chatbot understood the need for service options and provided a link to escalate the matter with repair experts.

```
Please type your question here ('q' to quit): My Mac is not working
Thanks for asking, here is the 1 most likely answer(s) from AppleSupport
We'd like to get you speaking to our experts here: https://t.co/IBIY3vMgPj
```

Figure 12. Sample chatbot session 2 (Apple, LDA)

Another question was asked regarding a Mac computer that was not working properly, and the chatbot provided a link to speak to the relevant experts. A crucial advantage here is that when further communication with human agents over direct message is required, the contextual information (the nature of the problem and the user's account details) can be carried over, instead of the human agent needing to clarify all these details. This saves time on both ends and allows the human agent to immediately discuss solutions tailored to the user's issue.

```
Is there anything else I can help?
Please type your question here or press 'q' to quit: q
Thank you for asking. Would you like to ask questions about other companies?
0 for Amazon, 1 for Apple, 2 for Uber, 'q' to quit: q
Thank you. Say safe and have a good one!
```

Figure 13. Ending a chatbot session

The chatbot prompts the user to enter additional concerns, until the user enters 'q' to end the process. The user is then asked about speaking with other companies before ending the session. Most users, who would likely have common problems that can be easily resolved, will be addressed at this stage. For issues requiring personal attention from human agents, contextual information can be carried over while the case is being escalated. This optimizes the company's customer service workflow and ensures customer satisfaction with an efficient issue resolution process.

Conclusion and Future Direction

In conclusion, we built a retrieval-based chatbot for customer service support of Amazon, Apple and Uber, by applying the Word2vec, TFIDF, LSI and LDA algorithms to select responses from historic replies from the customer service teams. By testing the four algorithms with the same input query, LSI performed better than the other three models. Performance was evaluated based on similarity scores and runtime, along with a qualitative analysis of the contents of chatbot responses (to check for relevance, tone and helpfulness).

Large companies are certainly interested in customers' concerns, which can be summarized by different topics. Topic modeling using LDA helps to identify a number of topics extracted from customers' tweets related to Amazon. Learning about the complaints or issues from the customers enables future improvement of the products or services and will improve customer satisfaction. On one hand, companies can put more energy into solving relevant problems; on the other hand, they can focus on the most salient topics and offer more detailed answers in those areas.

With regards to limitations, we only used one input question to compare the results of the models, which may not tell the whole story. A large number of customer queries are required to further evaluate the models and determine reasonable thresholds of similarity scores for different models. During the implementation phase, the chatbot was built using Tweets from 3 companies. Also, during the evaluation phase, Amazon and Apple were used as the main companies, and their circumstances may not translate equally to other companies. A more recent dataset could also be useful as the dataset used was created 3 years ago. Furthermore, since the twitter content includes irregular syntax and grammar, emojis and special characters, it was challenging to handle and preprocess the data. Another thing worth mentioning is that retweeting adds another dimension in terms of the frequency of the tweets. Therefore, using data from Amazon comments may offer better insights; as Amazon requires verification of members, it decreases scam users and improves the robustness of the dataset the models will be trained on.

References

[1] Rishi Sidhu. “How to Build a Chatbot — A Lesson in NLP” (2019).
[2] Soufyane Ayanouz, Boudhir Anouar Abdelhakim. “A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance” (2020).
[3] Daniela Ilieva. “Neuro-linguistic Programming Techniques for Perfecting Presentation Skills” (2014).
[4] Hussin Jose Hejase. “NEURO-LINGUISTIC PROGRAMMING (NLP): AWARENESS AND PRACTICE IN UAE” (2015).
[5] David M. Blei, Andrew Y. Ng. The Journal of Machine Learning Research. “Latent dirichlet allocation” (2003).
[6] Kaggle. “Customer Support on Twitter Dataset”.
https://www.kaggle.com/thoughtvector/customer-support-on-twitter (accessed 2021).