Providing Flow to Group Instant Messaging

Matthew Bu, Haoxin Luo, Alvin Leung
Natural Language Processing, Fall 2016
Rensselaer Polytechnic Institute
Troy, NY, USA

Abstract

This project attempts to solve communication problems that occur in text based conversations within a large group and to improve general communicative clarity and flow. Specifically, the research aims to use topic extraction and coreference resolution to provide an on-the-fly chat filter that helps chat participants focus on the current topics and archive old chat logs to be searchable by topic. Additionally, we aim to aid participants in expressing their emotions through the use of emoji by using sentiment analysis. Our findings show that it is possible to achieve non-disappointing accuracy using pre-trained models in part of speech, named entity, dependency, and coreference resolution to filter messages by topic.

1 Introduction

In any sufficiently large casual text group conversation, people frequently talk over each other. Consequently, participants not only write messages at the same time, but also talk about different topics simultaneously. This development of sub-conversations makes communication within the same chat room extremely difficult to follow. The most obvious and manual way of approaching this issue is for a user to create a new group thread or chat for every new subject being discussed. However, this method is both highly inconvenient and ineffective in that this creates barriers among groups of users and could still lead to more divergence of topics down the road.

Another source of inconvenience in group messaging is users tend to scroll through multiple pages of emoji (i.e. in Facebook Messenger) until they find the most accurate one that expresses their reaction to a recent message. Although popular messaging apps usually include a page of most recently used emoji, users would still have to spend time manually finding the appropriate ideogram, during which the group message may have already moved on to another subject.

With these drawbacks in mind, we propose creating an automated system that tags each message in a group chat for the topic that it contributes to. These tags can then be used to filter or highlight conversations of interest and even serve as a summary of what has transpired. Additionally, we expect to develop a system that can evaluate sentimentality of recent chats to help predict the emoji that the user might use, thereby expediting the emoji selection process. We envision this system will combine topic extraction and relation, coreference resolution, sentiment analysis, and wikification techniques to accurately create tags for each message. All of these areas of Natural Language Processing are well researched, and we hope to combine these well-established techniques to develop an elegant solution within a relatable user interface.

2 Related Works

Topic extraction is commonly done in two different ways: statistically and heuristically, and with machine learning methods. Supervised machine learning methods are simply impossible to train because there no annotated corpus of casual text conversations exist. On the other hand, Hasan et al. [3] have shown unsupervised machine learning methods such as TextRank, and PageRank performs poorly in cross-domain settings. Coincidentally, Hasan et al. [3] showed the statistical and heuristic method, term-frequency-inverse-document-frequency (Tf-Idf), is the best performing state-of-the-art method across the five domains they tested. As such, this paper focuses on a heuristic approach towards identifying key phrases. Roth [4] demonstrated that topic extraction and relation functions at an acceptable rate of 60 percent based off of term proximity, frequency, and lexical chain segments based on the work of Galley et al. [0] However, in all of these researches the authors were able to assume that there is only one topic for a given period of time, which is not the case in the context of group conversations. This assumption allowed frequency and machine learning methods to function because there is a large segment of text to analyze whether it be whole documents or just paragraphs. Another key distinction, especially between Roth [4] and this research, is that the problem this paper tackles requires us to tag all messages while Roth [4] only had to topic changes.

3 Approach

3.1 Topic extraction

We cannot assume that adjacent sentences comment on the same topic because they might be from different participants, and furthermore, any given member may partake in multiple sub-conversations. As a result, our topic extraction performs exclusively on a sentence by sentence basis and leaves sentences that do not specify a topic to the coreference resolution to tag. The topic extraction process takes into consideration parts of speech, named entities, and grammatical dependencies by leveraging the pre-trained Stanford's CoreNLP Java APIs. More specifically, the algorithm tokenizes each sentence then identifies and creates noun phrases from the tokenized inputs based on parts of speech tags. We also mark up the subject, object, main verb, and important modifiers in the sentence by using the dependency tree. Named entities and proper nouns are then normalized and disambiguated using data from Wikipedia while all other tokens are lemmatized. Due to time constraints, the wikification process simply takes the title of the most popular Wikipedia article returned by the search terms (the noun phrase). Each potential topic phrase is then given a confidence score, normalized to that of other phrases in the sentence, based on the combination of all the aforementioned metrics. The metrics are weighted based on perceived importance in the following order: named person, location, time, proper nouns, nouns, gerunds, object, main verb, and lastly subject. A combination of these properties could override a topic candidate that only fits in one of these categories. E.g. the sentence, “Anyone want to go to the concert at six?” might give *six* a score of 6 for being a named time/number, but *concert* would have a score of 8 because it is a proper noun and the object of the sentence. Thus the most likely topic for the sentence would be *concert* rather than *six*. This scoring method makes a generalizable confidence measure for single sentences. The top topic phrases are then returned as the topics of the given sentence(s).

3.2 Coreferencing and Filtering

The coreference resolution that comes with Stanford CoreNLP cannot be solely relied on for our situation to give even remotely accurate coreference results. This is due to multiple reasons, one of which is that in a group text conversation the perspectives are always changing from message to message. Yet there is no way to use the library to do coreferencing on single sentences or to adjust for the change of perspective. One major case with this situation is with the pronoun "I". Every instance of "I" would be assumed by the library to refer to the same speaker and hence every message containing the first person pronoun would be linked by coreferencing if we were to solely depend on it. Another difficulty is similar to how there is a problem in topic extraction due to user interleaving messages, coreferencing also cannot know whether any of the third person pronouns refer to a subject immediately prior. As messages are being sent by users, we need to classify them into different sub-conversations. In order to accomplish this, we calculate the probability that a message would be a part of a certain sub-conversation by examining the message metadata, topic similarity between the message and a sub-conversation, and coreference resolution between the message and sub-conversation.

Message metadata consists of information such the speaker of the message. In conversation, a person is more likely to continue a sub-conversation in which he/she was already speaking than to join a new sub-conversation. As such, messages from a specific user are given higher probability to join sub-conversations that he/she is already participating in.

Topic similarity between the new message and previous messages in a given sub-conversation was also used to calculate a probability of a message being a part of the sub-conversation. Using the topic extraction, we can get topic vectors consisting of topics and their confidence levels for the new message and previous messages. Topic similarity could then be calculated by using Euclidean distance between the two topic vectors where each topic is a new dimension with a value of its confidence level. We would then take the *cosine*[4] of the distance in order to give closer distances higher probability and farther distances lower probability.

Coreference resolution between message and sub-conversation also provided a probability that we could use to classify messages. Messages that provided coreference mentions to sentences in the sub-conversation can mean that it belongs in the given sub-conversation. Thus, we compared the coreference mentions of the sub-conversation with the message added to it, with the mentions of the sub-conversation alone to find all new mentions. We then calculated the probability by taking the total number of new mentions by the number of total mentions.

In order to now classify these messages, we weighed the previous discussed factors in the given order: topic similarity, coreference resolution, and message metadata. This ranking was determined by what we thought was most important in order to classify each message. Using the previously calculated probabilities and weight of each factor, we can now calculate a confidence level by taking the sum of products for each factor. We can then classify the message into the sub-conversation with the highest confidence level. However, sometimes, new messages can start a completely new topic. These messages would provide very low confidence levels for all previous sub-conversations, so we would classify them into new sub-conversation filters.

3.3 Emoji Prediction

Emoji prediction relies on performing sentiment analysis across the last five messages sent in a given group chat. From those previous messages, our analyzer gauges the general sentiment and outputs a list of emoji that are categorized under the evaluated sentiment. The sentiment analyzer is based off of Stanford's CoreNLP default model in their 2015-12-09 release, although the model could be easily retrained with custom input. CoreNLP also uses a Sentiment Treebank that can provide detailed evaluation for positive/negative classifications, contrastive conjunctions, and negations that help to predict fine-grained sentiment labels for all phrases to 80.7% accuracy, which was the kind of precision that we wanted for an instant messenger.

Once a sentiment value is returned, our application processes the raw value into a categorical enumeration type. For instance, a value of 2 returned from our sentiment analyzer would yield a *NEUTRAL*, or a value of 3 would produce *HAPPY*. The program then uses the enumeration to find a stored list of PNG filenames that represent all the emoji that were manually defined to depict happiness, sadness, neutrality, or so on.

Another feature that improves the accuracy of emoji prediction that is being developed is to use topic extraction and part-of-speech tagging to filter the emoji even further. For example, although the phrase, "I love Star Wars movies", would generate a list of relatively joyful ideograms, we want the application to be able to prioritize the heart emoji because the user specifically says "love". For this feature we would use our tagger to find certain target words that have a direct relational connection with an emoji, and use the closeness of that relation to produce a confidence score. Similarly in the example given, we would use topic extraction to output more specific Star Wars related emoji. Subsequently, we would show the list of predicted emoji ranked by highest confidence score first.

4 Experiments

4.1 Data

We ran our application against multiple datasets that were manually input with specifically defined subconversations to see if the software could catch and filter these divergent subjects. The datasets we tested against resembled the following format:

```
mattbu2###I really like president Obama.
haoluo###Me too, he is really cool!
haoluo###Anyone seen the latest Star Wars movie?
mattbu###I think Biden is cooler.
mattbu2###How old is Obama?
alvinleung###Obama is 55 years old.
mattbu###I've seen that movie!
mattbu2###I heard the movie's cast was epic.
haoluo###Nothing beats star wars!
mattbu###I agree, he is a great president.
alvinleung###too bad he's leaving office...
mattbu2###Obama is the best!!
mattbu###A new Star Wars movie is coming.
haoluo###When is the movie coming out?
mattbu###I don't know, probably this month.
```

Each line provides a username and the message sent by that user. For this example dataset, we expect the software to find at least two main subjects: Barack Obama and Star Wars. We intentionally included sentences that were ambiguous in order to see how our coreferencing algorithm would categorize those certain types of messages.

4.2 Evaluation

After running against the datasets, we counted the number of messages that were correctly categorized in the expected filter and divided that count by the total number of messages input to yield a percent accuracy. We also ran datasets that tested for sentiments and produced the following data:

Method	**Accuracy**
Topic Filter	63%
Sentiment	80%

5 Results

Overall, we were fairly impressed by the accuracy of our topic extractor and coreferencing resolutions. Our current model does harbor an underlying assumption that most of the given messages are in proper English, and although that is far from the case in real instant messaging, this was a step in the right direction for providing more flow to conversations. The sentiment analysis results proved to be reasonable as CoreNLP has a similar precision in terms of sentiment labeling. We even tested against sarcastic remarks that did not require too much outer context and found that most of the messages were evaluated correctly.

6 Future

Some considerations for future work include accounting for message frequency, incorporating proximity of messages (timestamp), removing stop words, using a relational database of terms to link non-dictionary synonyms, adjusting confidence scores, and improving disambiguation. We were also looking into integrating CMU's POS tagger which allows us to parse and use informal messages.

A potential improvement to the topic extraction method is to take into account the non-dictionary synonyms that occur within the context of each sentence and key phrase candidate. By building a categorical database of concepts, we can use the frequency and proximity of related phrases to create better topic tags by changing how specific the tag is. For example, in a conversation about movies, both *Star Wars* and *Doctor Strange* might show up and the current algorithm would tag each separately. Similarly in a conversation about the president, the actual topic could either be Obama or Trump. The first is an instance of the extracted topic being narrower than the topic, while the latter is when the topic is more specific than what our system will produce currently. In addition to helping change the specificity of the topic tags, an outside database can also help the coreference resolution model by providing the gender and plurality of non-common terms.

7 Team

All team members contributed to each other's work and lead tasks were assigned as follows:

Matt Bu, Undergraduate '18 – *User Interface, Sentiment Analysis/Emoji Prediction, Presentation*

Haoxin Luo, Undergraduate '18 – *Topic Extraction, Wiki API Integration*

Alvin Leung, Undergraduate '18 – *Coreference Resolution, Conversation Filtering*

References

[0]Galley et al.: Galley, M., McKeown, K., Fosler-Lussier, E., and Jing, H. 2003. Discourse segmentation of multi-party

conversation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (Sapporo, Japan, July 07 - 12, 2003). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 562-569.

Links

[1]Github:
https://github.com/mbu13/NLP_Final_Project

[2]Presentation:
https://github.com/mbu13/NLP_Final_Project/blob/master/NLP_FINAL.pdf

[3]Hasan et Al.
http://www.hlt.utdallas.edu/~vince/papers/coling10-keyphrase.pdf

[4]Roth:
http://nlp.stanford.edu/courses/cs224n/2010/reports/rothben.pdf

[5]CMU POS Tagging:
http://www.cs.cmu.edu/~ark/TweetNLP/#pos