Identifying Questions & Requests in Conversation

Kevin Quinn
University of Alberta
kfquinn@ualberta.ca

Osmar Zaiane
University of Alberta
zaiane@ualberta.ca

ABSTRACT

In building an automated conversation agent, that attempts to converse with a human with as human-like as possible manner, we require the agent to identify which dialogue act, or class, the sentence belongs to. Determining the sentence class of a spoken phrase is helpful in building an intelligent companion, because without it the response may seem out of place. In written language, sentences can be classified into three classes: *Declarative*, *Interrogative*, and *Imperative*. These classes indicate which dialogue act the sentence belongs to. What our system does is take spoken text, which contains no punctuation, and classify the text into the three aforementioned classes. In conversation, the type of spoken text can indicate the type of required response. Our system is able to classify spoken text with 82% accuracy on our semi-automatically constructed dataset.

1. INTRODUCTION

Around 41 million Americans are 65 and older, and they make up about 13% of the total U.S. population. Due to rising life expectancies world-wide, the population of seniors living alone at home is increasing. According to the U.S. Census Bureau projections, by 2050, one-in-five Americans will be 65 or older, and at least 400,000 will be 100 years or older [2]. Not all will be living assisted in collective dwellings. A large proportion will be living in their own homes, and very often alone. In 2011, the Census of Population in Canada counted nearly 5 million seniors aged 65 and over of which 92.1% lived in private households [15]. The benefits of companionship for the elderly to maintain a good state of their physical and mental health has been demonstrated [11] and many seniors' companionship services exist. However, these services are not always available and certainly do not scale with the expected significant rise of seniors' population. One solution is a software agent that could converse intelligently and could be embedded in tablets or other domestic appliances.

We have built a prototype conversation agent named ANA (Automated Nursing Agent) that attempts to converse with the elderly. The intent is not to challenge the Turing Test [5] but to create an adaptive conversational agent that can generate and exchange pleasing fluent dialogues for the elderly. ANA not only answers questions, reminds the interlocutor of occurrences such as taking medicine or going to an event, but also generates questions or personalized statements. This is done by dynamically creating a personalized knowledge base about the interlocutor and exploiting this knowledge base during the dialogue. ANA's input is text via a speech-to-text module, and the output is also text that is converted by means of a text-to-speech module. For ANA, identifying the sentence class in the input is crucial in order to determine the following action such as answering a question, populating the knowledge-base or generating a statement response.

Much work has been done in the field of speech act identification (SAI). This field attempts to determine the function of a particular utterance. These functions can vary depending on the domain needed. An example of this would be in emails it is necessary to know when a person is requesting the help of another person. In this scenario the function we would want to identify are requests or commands. Another example would be trying to identify questions or inquiries. The paper by [9] attempts to extract questions from Twitter. There are many different ways to categorize sentences and in this paper we assume there are three types: declarative (statements), interrogative (questions), and imperative (requests).

We have created a conversational agent named ANA that converses with a human. ANA's goal is to communicate with the elderly and improve the quality of their life. This is similar to the Chester project from Allen et al. which also uses conversation to assist users [1]. They focus more on providing the user with information about their medication.

To make an intelligible response ANA must be able to classify the input into either a question, request, or statement. If a question is detected ANA will attempt to answer that question. Potential questions could be related to the news, weather, or about the user's family. Similarly if a request is detected ANA will attempt to fulfil the request. This could mean alerting the nurse, opening the window, or calling a family member. It is important to identify the type of input, because this will lead to a better response. Applications such as Google Now or Siri use a method to distinguish between questions, commands, and statements.

There are many problems in determining the function of a sentence. Our primary assumption is that we do not have access to the proper punctuation. Without punctuation this task becomes much more difficult. Usually if the sentence ends with a question mark we can assume it is a question. However the data in which we run our experiments does not have access to this.

C3S2E '14 August 04 - 07 2014, Montreal, Canada

http://dx.doi.org/10.1145/2641483.2641534.

In this paper we assume that a sentence can only belong to one class. This is not always true in practice. For example, "Do you have the time?". This can be a request (imperative) or question (interrogative).

Another aspect that is difficult to detect is the intonation of the sentence. The way a sentence is intonated can determine the class even if the content is identical. Take almost any simple sentence, such as "I like fish", change the intonation, and it will become a different class.

You walk to school everyday. (dec)

You walk to school everyday? (int)

You walk to school everyday. (imp)

There are times when this is impossible to decipher. However, there are still distinguishing features among the three classes that can aid us in our task.

We propose a support vector machine with hand crafted features for this task. We will use part of speech (POS) ngrams, word ngrams, parsing, and word clusters to model the data. We believe we are the first to use word cluster features for speech act identification.

This paper solely focuses on the problem of sentence function identification. The aspects related to knowledge base construction and building an intelligent agent are not addressed here in.

In this paper we will first discuss previous work regarding identifying sentence function. We then elaborate on our proposed method and the experiments. Finally we conclude with a discussion of our results, and hints for future work.

2. RELATED WORK

For the task of speech act identification a sentence is classified in different speech acts. Various work has been done in the field of question detection in twitter, email, and online discussion boards. Requests detection has been applied to the email and message board domains.

Searle published various works in the field of speech acts that provides a solid foundation language and dialogue [12] [13]. He discusses philosophical ideas associated with language and introduces a taxonomy of speech act classes.

Stolcke et al. implement a system that uses statistical methods to model discourse structure for conversational speech [16]. They defined 42 'dialog' acts, such as yes-no questions or apologies, and labelled the switchboard telephone dataset with these acts. They focus on using statistical models such as a hidden markov model to calculate discourse probabilities. For example, if the previous sentence was a question what is the probability of the current sentence being a question.

A few papers have attempted to label emails with speech acts. The main idea here is to identify "intent" or "purpose". Cohen et al. discuss identifying requests for meetings or sentences that provide information [3]. They have different classes such as requests, commitments, proposals, and reminders. To identify these classes they implement a support vector machine with features such as ngrams and part of speech phrases. Similarly [8]'s paper also attempt to find actions within emails. They also use a support vector machine with various features to accomplish this task.

There was a semi-supervised approach to speech act identification from [6] that used sub tree pattern mining. They had classes such as statements, rejections, yes-no-questions, wh-word-questions, or uncertain responses. Their work was done on forum and email data.

The work by [10] attempts to identify speech acts within message board posts. The type of language used is similar to what we would like to use in ANA. They try to extract four different speech acts from the first post: commissives, directives, expressives, and representatives. The key difference is that they assume a message can contain multiple speech acts where as we attempt to classify the entire sentence. They implement an support vector machine (SVM) classifier with various hand crafted features to solve this problem.

In [9] they attempt to identify questions within tweets. The first task is to find tweets that contain questions. For this they use the Prefix Span algorithm which attempts to find frequent sub sequences or sets of words to use as features. Next they use both a unsupervised and supervised approach with various features to extract the question from the tweet. This paper also has to deal with noisy text data that is frequent within tweets.

[14] attempt to detect questions and answers from emails. They also use a machine learning approach with discriminative POS bigrams combined with beginning ngrams and ending ngrams. [4] also attempts to detect question/answer pairs, but their domain is in forum posts. They attempt to leverage labelled sequential patterns which combine POS tags and words.

All of the previous works have different definitions for speech acts. They choose the classes that suit their task and our work is no different. Most of the methods use both patterns and supervised learning. The data used is a combination of informal and formal language.

3. METHOD

We previously discussed how various classes for speech act identification are defined depending on the task. We decided to use the classes: declarative, interrogative, and imperative because they fit into ANA's definition of a potential user utterance. The imperative necessitating assistance in execution of commands; interrogative requesting an answer to a specific question via a search on the Internet or a query to the personalized knowledge base; and declarative providing an opportunity to extract information for the knowledge base or simply enticing a response for a dialogue.

3.1 Speech Act Definition

Declarative: A declarative sentence states a fact. These sentences occur the most in articles, book, and news. Each sentence has a definitive subject and object. They attempt to relay information or an idea. It is difficult to determine distinctive features of these sentences because they vary tremendously. Usually we can identify them based on their punctuation, but we would like to find alternative ways to detect them.

Usually in conversation declaratives state facts about the individuals such as "I like cake" or "I am 25 years old". It is possible for a user to state a fact similar to a news article.

Interrogative: An interrogative sentence asks a question. Normally these end with a question mark in written text, but in our case they are absent. There are many aspects that make a sentence interrogative. They usually begin with a wh-word such as who, what, when, etc. They can also end with tag phrases such as “don’t you" or “right". The person speaking usually wants information from the listener. This could be a new piece of information or a clarification on a previous statement.

Imperative: An imperative sentence is a request or command. These are common in driving directions, access to functionalities on a smart phone, cooking recipes, and explanations on how to finish a task. They also occur in spoken text when the speaker wants the listener to help them somehow. They can even be as simple as one word such as “move" or “go". In our context most imperatives are between two people.

Function	Example
Declarative	*I like to eat cheese cake.*
Interrogative	*What does Phil like to eat?*
Imperative	*Open the window please.*

Table 1: Examples of Sentences and their Class

3.2 Features

In this section we will discuss the features we opted to use. Most of these features are very popular in the current literature such as lexical and syntactic features. However we have not seen other attempts to leverage dependency parsing or word clusters for the speech act identification task. We will now briefly describe the features we use.

Discriminative POS Ngrams: The idea here is that many of these sentences contain a certain grammar structure that could potentially identify them. This is why we chose to use POS ngrams. We went through each class of sentence and found all frequent POS bigrams. Next we needed to eliminate the bigrams that were not frequent which meant the bigrams that appeared less than three times. This was to make sure that a bigram was important to a particular class. We found that bigrams and trigrams were the most effective.

The reason we wanted to use POS ngrams was because various phrases within the same class have similar part of speech patterns. Here are two examples: “Open the window", “Call the Doctor". They both possess the same POS representation “VB DET NN".

Ngrams: There are phrases and sequences of words that are very influential in certain sentences. We evaluate different unigrams, bigrams, and trigrams with pointwise mutual information to see if they would be valuable to use as features. The motivation behind this feature set is that there are phrases that are specific to one class. Tag questions usually end with a similar phrase. “You like pizza, right?" or “You like pizza, don’t you?". Again we used both bigrams and trigrams in our task.

Dependency Links: We suspect that parsing could potentially be useful in our task. Certain dependency links are present in sentences that determine which function it is. We consider pairs of links as features in this work. How exactly are these extracted? First we find the root of the sentence and extract all of the outgoing links. Second we find all of the dependent nodes and find all of the outgoing links of those nodes. So for one sentence we will extract multiple pairs of dependency links. Take the link pair “nsubj → det". This link pair occurs very frequently within declarative statements such as “The concert begins in two hours.". In Figure 1 the extracted pairs would be “nsubj det” and “prep_in num”.

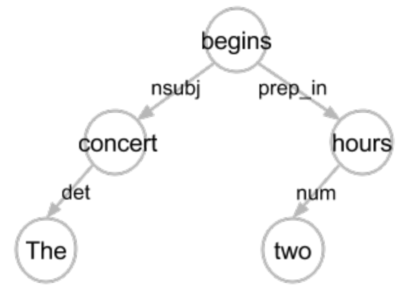

Figure 1: Dependency link example.

Constituency Parsing: We would also like to use constituent parsing in our task. We believe that constituent parsing would be useful in detecting interrogative sentences.

```
(ROOT
 (SBARQ
  (WHNP (WDT What) (NN time))
  (SQ (VBZ is)
   (NP (PRP it)))
  (. .)))
```

In this example the constituent parser includes the “Q" label after the root node. This is what we are looking for and we are assuming that it will help label interrogative sentences.

Word Clusters: In the paper by [18] they evaluate various corpera using word clusters to improve supervised learning. They evaluate the Brown clusters, Collobert and Weston (2008) embeddings, and HLBL (Mnih & Hinton, 2009) embeddings of words on various NLP tasks. We believe that using these clusters could improve the performance of our task. Common imperative verbs such as open and close will be clustered together and could help identify unusual syntactic phrases.

For a given sentence we iterate through every word and determine which cluster each word belongs to. A sentence can have multiple word clusters as features. We use a stop word filter to remove words that are too common or semantically unimportant. In order to utilize the clusters we need to decide how many clusters should be used. We evaluated the 100, 320, 1000, and 3200 class clusters in Table 4.

3.3 Classifiers

In order to create our classifier we used the svmlight library [7] in conjunction with the Java layer from [17]. We created three binary support vector machines, one for each class. We used a linear kernel with the default settings supplied from Theobold’s code. We assumed that because the data was balanced and the features were the same for each classifier, that we could take the label from the classifier with the largest objective function value.

4. EXPERIMENTS

4.1 Data

Our experiments are run on data constructed both manually and automatically. We had access to approximately 400 self-annotated sentences from our ANA project. This was constructed manually by one annotator. These sentences consist mainly of short and simple sentences. They are mostly in the context of a conversation. This means they contain informal speech and pronouns. All of this data is synthetic, noise-free, and simple.

Speech Act	# Sentences	Distribution
Declarative	350	33%
Interrogative	350	33%
Imperative	350	33%

Table 2: Class Distribution

In order to supplement this data we added more by gathering sentences from various sites. Our assumption was that by extracting sentences from a particular site could determine the label. For instance, the site “uselessfacts.net" which contains various facts and statements. Therefore we could assume that every sentence extracted belonged to the declarative class. Similarly we extracted articles from “wikihow.com" that instructed the user how to complete certain tasks. Again our assumption was that sentences extracted here could safely be labelled as imperative. Note that when extracting sentences from wikihow we only used the first sentence. This sentence was extracted by using an HTML parser to find and extract a particular class tag. In order to extract interrogative sentences we relied on punctuation. We parsed sources such as news articles, the enron email dataset, reviews, transcriptions and forums discussions. From these we extracted the sentences which contained a question mark. The sentences contained can be quite noisy when compared with the ANA dataset. We used these sources to construct a dataset of 1050 sentences. All ending punctuation was replaced with an “end" tag. Our assumption is that we do not have access to the punctuation. Of this data we used 1044 sentences for training and 232 for testing.

4.2 Baselines

In order to determine whether our classifiers are performing well we implemented two baselines to compare against. The first baseline is simply choosing a random class for a given sentence. Each class is weighted equally.

The second baseline implemented is an unsupervised rule based approach. We went through the dataset and attempted to find frequent patterns for each class. These patterns included certain words, part of speech fragments, and phrases. The position of these patterns were also important. For example if the sentence ends with “right" or “eh" or the sentence begins with a verb. Below is a list of the patterns used:

- Sentence begins with a verb.
- Sentence ends with "tag" word (eh, right, really, etc).
- Presence of certain interrogative phrases such as (don’t you, do you, isn’t it, can’t he, etc).
- Sentence begins with “wh-word".
- Presence of emotion or sentiment words.
- Presence of request verbs such as (can, pass, give, will, etc).

Our hypothesis is that our classifiers should out perform both of these baselines.

4.3 Metrics

Traditionally in information retrieval the metrics used are a combination of precision, recall, and f-score. For our task we use: per class precision (P), recall (R), f-score (F), and total accuracy (A). We calculate the precision, recall, and f-score for each binary classifier in addition to the overall accuracy of our classifier, and this evaluated on combinations of our selected features.

5. RESULTS

In order to evaluate which cluster size was optimal we re-ran our experiments on the last feature set (POS+WRD+Clus) with different cluster sizes.

Cluster Size	Accuracy	Average F-Score
100	79	79%
320	81	80%
1000	82	81%
3200	67	74%

Table 4: Cluster Size Results

5.1 Performance

The cluster size results shown in Table 4 show that the optimal number of clusters should be 1000. As the number of cluster increases the performance seems to increase, but after using 3200 clusters the performance drops considerably.

From our results in Table 3 we can see that our SVM approach outperforms both the random and rule-based approaches. The rule-based approach simply does not offer much, because the patterns contain many false-positives. With regard to each class the imperatives achieve the highest f-score.

Because our task is slightly different than other systems it is difficult to determine how good the results are. That said there are components of our task that are very similar. Much of the related work deals with question detection which is essentially the interrogative class. The paper by [4] achieves an f-score of 97% on question detection with the use of labelled sequential patterns. This is most likely because their data is much more formal than ours.

Another paper that we can make a few comparisons with is [10]. They extract speech acts from message board posts. One of their classes is called a *Directive* which is a combination of a question, request, or invitation. They are able to achieve a 86% f-score.

5.2 Error Analysis

As stated before a sentence can be classified entirely on how it is intonated. The sentence “I like fish", can be simultaneously a declarative sentence and interrogative sentence. This is because the emphasis is missing from pure text. Unfortunately there is no way to distinguish between these without the vocal information.

In our dataset there were numerous one or two word sentences. These are tricky because they offer little in the way of context. If given the sentences “yeah", “right", “move", or “yes" out of context

	Declarative			Interrogative			Imperative			
	P	R	F	P	R	F	P	R	F	A
Random	.25	.14	.18	.31	.30	.30	.41	.34	.37	.24
Rule-based	.39	.90	.55	.83	.13	.23	.91	.58	.71	.53
POS	.69	.83	.76	.81	.66	.72	.83	.88	.85	.77
POS+WRD	.72	.86	**.78**	.81	.63	.71	.86	.88	.87	.80
POS+WRD+Lnks	.70	.82	.75	.76	.63	.69	.86	.87	.87	.78
POS+WRD+Lnks+Clus	.75	.83	**.78**	.85	.70	.77	.84	.92	.87	.81
POS+WRD+Clus	.75	.83	**.78**	.88	.70	**.78**	.86	.92	**.89**	**.82**

Table 3: Results

could a human classify them into their correct categories? It might be useful to consider the previous sentence or sentences in this case.

The dependency parsing was quite disappointing. It appears that pairs of links were only occasionally useful, and mostly detrimental in classifying sentences. There were a few instances where the parsing did help. The sentence "After the snow storm the air smelled fresh and clean." is originally labelled as an imperative sentence mostly because of the POS "det nn", but because the sentence contains the "nsubj $\rightarrow$ det" link it is labelled correctly as a declarative statement.

	Pred Dec	**Pred Int**	**Pred Imp**
Gold Dec	59	5	7
Gold Int	**17**	53	6
Gold Imp	5	2	78

Table 5: Confusion Matrix.

Table 5 shows where the primary problems lies. When attempting to predict the value of an interrogative sentence our system is choosing instead to predict declarative. Instances beginning with "You" such as "You know" or "You are, ... right" are where our system fails consistently. Other issues lie with the intonation problem. For example, the sentence "she enjoyed them" can be both declarative and interrogative depending on factors other than text.

6. CONCLUSION & FUTURE WORK

Our task was to classify sentences according to their function. We defined these classes as *Declarative*, *Interrogative* and *Imperative*. We have presented a machine learning approach using an SVM with various NLP features that achieves good results on a varied dataset. Our main problem is the lack of external context such as sound or previous conversation.

For future work we will consider hybrid classes such as interrogative-imperatives sentences. Many sentences do not fall nicely into one of our three classes, and it would be useful to identify these. An example is "Do you have the time", where the person is asking for the time, but they are also asking a yes/no question. The response of a conversational agent could be improved significantly with this implemented correctly.

Another area of improvement could be evaluating whether identifying the class of a sentence can improve the response of a conversational agent. The primary assumption of this paper is knowing the class of a sentence can improve the quality of the response, but this is not 100% obvious. A user study containing possible responses and sentence class would help prove this.

In this paper we consider each sentence separate and without conversational context. We believe that including the previous sentence or set of sentences could improve the overall result. A hidden markov model could model the probabilities of one sentence type following another similar to what was done by Stolcke et al [16]. This would especially be useful when the sentence itself is very short.

7. REFERENCES

[1] J. Allen, G. Ferguson, N. Blaylock, D. Byron, N. Chambers, M. Dzikovska, L. Galescu, and M. Swift. Chester: towards a personal medication advisor. *Journal of Biomedical Informatics*, 39(5):500–513, 2006.

[2] P. D. Census Bureau. Projections of the Population by Selected Age Groups and Sex for the United States. `http://www.census.gov/population/projections/data/national/2012/summarytables.html`, 2012. [Online; accessed 30-January-2013].

[3] W. W. Cohen, V. R. Carvalho, and T. M. Mitchell. Learning to classify email into"speech acts". In *EMNLP*, pages 309–316, 2004.

[4] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474. ACM, 2008.

[5] B. J. Copeland. The essential turing. the ideas that gave birth to the computer age. *AMC*, 10:12, 2004.

[6] M. Jeong, C.-Y. Lin, and G. G. Lee. Semi-supervised speech act recognition in emails and forums. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1250–1259. Association for Computational Linguistics, 2009.

[7] T. Joachims. Making large scale svm learning practical. 1999.

[8] A. Lampert, R. Dale, and C. Paris. Detecting emails containing requests for action. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 984–992. Association for Computational Linguistics, 2010.

[9] B. Li, X. Si, M. R. Lyu, I. King, and E. Y. Chang. Question identification on twitter. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2477–2480. ACM, 2011.

[10] A. Qadir and E. Riloff. Classifying sentences as speech acts in message board posts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 748–758. Association for Computational Linguistics, 2011.

[11] P. Raina, D. Waltner-Toews, B. Bonnett, C. Woodward, and T. Abernathy. Influence of companion animals on the physical and psychological health of older people: an analysis of a one-year longitudinal study. *Journal of the American Geriatrics Society*, 1999.

[12] J. R. Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.

[13] J. R. Searle, F. Kiefer, and M. Bierwisch. *Speech act theory and pragmatics*, volume 10. D. Reidel Dordrecht, 1980.

[14] L. Shrestha and K. McKeown. Detection of question-answer pairs in email conversations. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[15] C. Statistics. Living arrangements of Seniors. `http://www12.statcan.ca/census-recensement/2011/as-sa/98-312-x/98-312-x2011003_4-eng.cfm`, 2013. [Online; accessed 30-January-2013].

[16] A. Stolcke, E. Shriberg, R. Bates, N. Coccaro, D. Jurafsky, R. Martin, M. Meteer, K. Ries, P. Taylor, C. Van Ess-Dykema, et al. Dialog act modeling for conversational speech. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105, 1998.

[17] M. Theobald. SVMLight Java Layer. `http://adrem.ua.ac.be/~tmartin/`, 1999. [Online; accessed 30-January-2013].

[18] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, 2010.