Capturing an utterance's dialogue act

Dana Doytch

cl2ddoyt@cling.gu.se

Essay in Pragmatic XD6100 5p, B-level
Computational Linguistics, Department of Linguistics
Gothenburg University

10 April 2005

Supervisor: Torbjörn Lager

Contents

1. Introduction

1.1 Dialogue Acts

A dialogue consists of a sequence of utterances. These utterances are a kind of speaker actions, or speech acts (Austin 1962). One type of speech act is called *illocutionary act*: the act of asking, answering, promising, etc. in uttering a sentence. The uttered sentence is a contiguous series of words from a given speaker, which serves a precise function in the dialogue (or sometimes more than one), i.e. every utterance is explainable in relation to an activity.

Dialogue Acts are realized in speech using utterances. They mark important characteristics of utterances, indicating the role or intention of an utterance in a specific dialogue. A fragment of a labeled dialogue is illustrated in table 1.

Speaker	Utterance	Dialogue Act
A1	I have some problems with my homework.	Inform
A2	Can you help me?	Request
B1	I can't help you now	Reject
B2	Let's discuss it Friday…..	Suggest
A3	Okay	Accept

Table 1: a sample dialogue labeled with dialogue acts.

1.2 Dialogue Acts classification problem

Ambiguity is a crucial problem for natural language understanding/processing. In dialogues, a surface utterance may be interpreted differently depending on its context, for example: utterance A2 in table 1, looks like a question but is interpreted as a request.
Dialogue act classification problem can define as follows: "determine the dialogue act of an utterance (contextual words), given the contextual words".
We want to assign the appropriate dialogue act to each utterance in a dialogue in order to reach an accurate dialogue act classification.
To solve the dialogue acts classification problem we need to understand, by analyzing the context, the role the user's utterance plays in the dialogue.

1.3 Purpose

This paper presents an attempt to improve the performance of dialogue act tagging by investigating how the listener manages to interpret a conversational partner in a dialogue.
My purpose is to investigate the use of the fine aspects of the English language in order to help predict utterance types.

The method I chose to perform this task with is by allowing a number of informants to identify dialogue acts in the written dialogues which I chose to pick from the MapTask corpus[1].

I hope to find a pattern in these identifications that will help to improve the result of automatic dialogue act classification.

2. Background

2.1 Machine learning

The field of machine learning is concerned with the question of how to construct computer programs that automatically learn with experience.
Learning: everything that have a pattern (sequence) and that we have enough information about, we can learn!
A machine learns whenever it changes its structure, program or data (based on it's inputs or in response to external information) in such manner that it's expected future performance improves.

Along with the growing flood of online annotated data (corpuses) and recent progress in algorithms and theories, machine learning has become more interesting tool in the area of Natural Language Processing on issues like Part-of-Speech Tagging, Word Sense Disambiguation, PP-attachment Disambiguation, Text and Speech Categorization, Dialogue Act tagging etc.

Some of the well known machine learning techniques used today include:
-*Inductive (symbolic) learning algorithms*, learning strategies as rote learning, learning by being told, learning by analogy, learning from examples, and learning from discovery. Knowledge is represented in the form of symbolic descriptions of the learned concepts, e.g., production rules or concept hierarchies.
-*Evolution-based genetic algorithms,* strategies emphasize individual behavioral changes based on the principle of genetics.
-*Multiple-layered, feed-forward neural networks*, knowledge is learned and remembered by a network of interconnected neurons, weighted synapses, and threshold logic units.

[1] see 3.1

2.2 Machine Learning for NLP

Classification is well-studied machine learning task. The task is to find the class for which an instance belongs to.

Since the early 90's, machine learning algorithms have been successfully applied to classification tasks and recently there have been significant advances in machine learning approaches for developing several models to automatically detect dialog acts from transcribed or automatically recognized utterance. One of the most popular approaches is Transformation-Based Learning (TBL).

2.2.1 TBL -Transformation-Based-Learning

Transformation-Based Learning is a machine learning technique devised by Eric Brill 1995, based on the idea of successively transforming data to correct errors that gives the highest error rate. It is very flexible and powerful learning algorithm that achieved excellent result on part-on-speech-tagging[2].

The learning algorithm:
Learning process: initial tag sequence is build for the annotated training data. Iteratively, all possible rules are tested and compared to known tags. The rule that best repairs the current errors is applied to the sequence and added to the ordered list of transformations.
Tagging process: un-annotated text is passed through an initial-state annotator which applies most frequent tag for every word. The tag sequence is iteratively refined by applying transformation rules in rank order.

The algorithm can exploit a wide range of lexical and syntactic regularities via transformation rules that it iteratively acquires. The transformation rules obtained are usually few and meaningful. They are acquired by instantiation of a predefined set of template rules which is based on theories and assumptions. The template rules serves a learned model which contains the relevant features and the important relationship between features and tags. The generated output gives an overview for the theories to explain the data, this allows developers to easily understand, manipulate and debug the resulting system.

The efficiency of this algorithm has proved to be low, it takes long time to train the data and the training memory usage is high.
However there are some alternative machine learning systems which provide an extensions of TBL i.e. LazyTBL(Samuel 98), μ-TBL(Lager 99), FastTBL (Ngai & Florian 01). Some of the goals that these alternative methods have, are to make the TBL system robust, accurate and domain-independent.

---

[2] Samuel et al. [7.6] describes a useful analogy for understanding the TBL paradigm.

2.2.2 Dialogue Act Tagging

Automatic interpretation of dialogue acts, called Dialogue Act Tagging. The Dialogue Act Tagging task is to determine the dialogue act of an utterance by labeling each utterance in a dialogue with its proper dialogue act.

An attempt to apply Transformation Based Learning to discourse-level problems, made by Samuel[3], has proved that machine learning can be an effective tool for identifying dialogue acts. Samuel's approach utilizes transformation based learning over a number of utterance features, including utterance length, speaker turn and dialogue act tags of adjacent utterances.

A similar attempt, based on Samuel's experiment was made by Torbjörn Lager and Natalia Zinovjeva[4], using the μ-TBL system[5].
Lager and Zinovjeva uses dialogue models trained on previous moves, actual word/words, speaker turn, speaker role and utterance length, to predict the current move type.

The language models, described in these attempts, do not take into account regularities in sequences of dialogue acts at different points in the conversation and the result reported may be improved with the help of human knowledge. This can be achieved by improving the set of templates so that the machine learning system will generate a correct sequence of transformation rules and provide high accuracy when evaluating the tagger.

With the knowledge taken from these experiments I decided to investigate the possibility of improving the performance of Dialogue Act Tagging, by rewriting the current set of templates.

I believe I can improve the result of Dialogue Act Tagging by writing a useful set of templates that highlights the conditions and combinations of conditions that are relevant for determine the correct dialogue act tied to an utterance, based not only on syntactic and semantic but also on pragmatic knowledge.

---

[3] see 7.5
[4] see 7.2
[5] The μ-TBL system, programmed by Torbjörn Lager, is implemented in Prolog and is based on the original TBL. http://www.ling.gu.se/~lager/mutbl.html

3. Method & Material

3.1 The corpus data

The experiment reported here use a subset of the HCRC MapTask corpus[6].
The MapTask is a corpus of spontaneous goal-directed dialogue speech, which consists of a total of 128 task-oriented dialogues. The MapTask corpus has a limited vocabulary and structured speaker roles, each conversation has two participants each with different role called the giver and follower. Generally the giver is giving instructions and guiding the follower through the route on the map. The corpus has been analyzed using the game move theory modified for Maptask dialogues.
The coding schema which I use in my experiment corresponds to the twelve dialogue act moves included in the corpus coding schema[7].

3.2 Elicitation Method

A number of randomly picked dialogues were chosen from the MapTask corpus, mainly because I intend to test the result driven from my experiment by using the same test and training data which Lager and Zinovjeva used in their experiment and partly because this task oriented corpus is easy to analyze.
A small number of informants were asked to classify each utterance in the given dialogue with one of the 12 act moves and highlight the features that made them classify the utterance the way they did.
The complexity of the dialogues and the difficulties involved in finding the correct dialogue act that matches with the tagged corpus, resulted in the two following experiments:
Experiment 1: According to the Dialogue Act model used, conversations are viewed as a series of games and on the base of finding the exact move made by player, the participants were asked to pretend playing the actual game while examining the dialogue.
Experiment 2: The participants were shown the correct dialogue act classified to each utterance and asked to try to identify the reason to why the utterances were classified the way they are.

[6] The HCRC Map Task Corpus, see 7.7
[7] see 8.1

4. Elicitation Result

As a result of my experiment I found common features that the informants mark as their hallmark. The actual word and word sequence that made them interpret an utterance the way they did is summarized in table 2.

Act	Captured by	Example
query_w	Words such as "where"," which"	A: "**where** is that?"
reply_w	Words like "right", "really", when the previous move is a *query_w*	A: "where is that?" B: "**eh,is at the top…right hand side**."
explain	Sequence of words "I know", "I've", "that's", "it's" followed by a *reply_y* or *a reply_n* move.	A: "no" A: "**I've** got a …"
instruct	Words as "imagine", "you have to", "you're".	A: "and **you're** going around"
check	Questions which includes the words "have you"," have I", "to the", "I should". When the previous move is *instruct*.	A: "about three inches…you're in the" B: "so **I should** be around the middle?**"**
query_yn	Sequences such as "do you..", "have you", "is that", "is that on".	A: "**have you** got a walked city"
acknowledge	Words that appear in the beginning or at the end of an utterance, "right", "uh-huh", often after a *reply* move. Single words as "uh-huh" when the previous move is *instruct*.	B: **"Right** I see**."** B: "**Right.**" A: "right just curve around it." B: "**uh-huh**."
align	"right" at the end of an utterance. The utterance is formed as a question, when the previous move is *explain* or *instruct*.	A: "imagine there is a graveyard" A: "**right**?"
clarify	words from the previous utterance were repeated	A: "you must **pass** the diamond mine" B: "do I have to **pass**….**pass** the …"
ready	Utterances begin with "so", "right", it may appear before a *check* or an *explain* move.	A: "**right so**," A:"have you got…"

Table 2: summarizing the way the acts were captured.

Some observations drawn from the experiment:
In some parts of the dialogue difficulties where found distinguishing between the moves *acknowledge* and *clarify*, since the repeated words from the previous utterance are not always an indication for the *clarify* move.
There are some similarities in the words that indicates *explain* and *instruct* move, both moves often appear after the *acknowledge* move and the length of these moves is sometimes equal.

A move sequence such as *explain* followed by *acknowledge* is common near the end of a game as the goal of that game is achieved.

The appearances of the word "uh-huh" at the beginning of an utterance can be interpreted both as *reply* and *acknowledge* moves. It is however most likely to appear as a *reply* move when followed by the *query_yn* move.

The indicator for the *check* move is not always clear and seam to vary extensively in the different dialogues.

A *ready* move at the beginning of a dialogue may be more emphatic than one in the middle of the dialogue.

Furthermore all the informants marked that words "no" and "yes" as the indicators for the moves *reply_n* respective *reply_y*, which is evident.

My observations resulted in some of the following templates:

query_w template 2 – classifies the current utterance as *query_w* if question-words like: "where", "who", "how" appears in the first position of the utterance.

reply_w template 8 - classifies the current utterance as *reply_w* if the previous utterance or the pre-previous utterance is classified as *query_w* and the current utterance consists of more then three words.

explain template 11 - classifies the current utterance as *explain* if the current utterance contain a word like "I've" and the previous utterance is classified as *reply_y*. template 12 - classifies the current utterance as *explain* if the previous utterance was not uttered by the same speaker (the speaker's role has changed) and the previous utterance is classified as *reply_y* or *reply_n*.

instruct template 14 - classifies the current utterance as *instruct* if the current utterance consists of more then ten words and the previous dialogue act is classified as acknowledge act.

check template 6 - classifies the current utterance as *check* if this utterance is uttered by the same speaker as in the previous one (the speaker's role didn't changed) and previous utterance is classified as *instruct*.

template 10 - classifies the current utterance as *check* if the first word of the utterance is "so", "you've".. and the current utterance consists of more then three words.

query_yn template 4 - classifies the current utterance as *query_yn* if it contain word sequences like "have you".

align template 5 - classifies the current utterance as *align* if the previous dialogue act is classified as *instruct* or *explain.*

clarify template 9 - classifies the current utterance as *clarify* if a certain word in the current utterance is repeated.

ready template 13 - classifies the current utterance as *ready* if words like "so", "right" appears in the current utterance and the utterance length is longer then ten.

5. Machine learning experiment

As mentioned above, one way of improving the efficiency of a transformation- based learner system is by rewriting the rule templates, manually developed in advance. Since the set of templates limit the set of transformations, it is necessary that the choice of these is made carefully.

Following the observations driven from my experiment (represented in table2), I designed a set of templates with a total sequence of 14 rules. Using the same test and training data which was used in Lager and Zinovjeva experiment, I ran the test based on the preliminary result. The result of this test is given in table 3.

	Lager & Zinovjeva result	My result
Number of templates	16	14
Number of rules learned	211	220
Initial test data accuracy	19.9% (482 errors)	19.9% (482 errors)
Final test data accuracy	62.0% (229 errors)	64.5% (214 errors)

Table 3: the result of the test.

As shown in table 3, the learning process resulted in a sequence of 220 rules, the tagging result was improved with 2.5% which is not much but indeed an increase in correct percentage and a decrease in error occurrences.
Two examples from the rules found by the μ-TBL system:

- *da:instruct>check <- u_first:'To'@[0] & u_length:'>3'@[0].*

This rule is an instance of template 10, it can be explained as follows: "replace the classification tag *instruct* with the classification tag *check* if the first word in the current utterance is 'To' and the length of the utterance is longer then 3 (consists of more then three words)."

- *da:instruct>reply_w <- da:query_w@[-1,-2] & u_length:'>3'@[0].*

This rule is an instance of template 8, it can be explained as follows: "replace the classification tag *instruct* with the classification tag *reply_w* if the previous utterance or the pre previous utterance is tagged with *query_w* and the length of the current utterance is longer then 3 (consists of more then three words)."

There are three stages to how the rules are learned:

1. The system labels every utterance with its most likely dialogue act tag.
2. It examines every possible transformation rule that results in the most improved tagging and iteratively tests all possible instantiation of it.
3. When there are no more rules to apply, it retags the data according to rules selected.

6. Summery and Conclusion

In my experiment I examined the possible features that make us recognize a dialogue act. With the help of a limited number of informants I could summarize some of the features that appear to be common when identifying a dialogue act.

Based on these result I constructed a small set of templates which gave a fairly good result when tagging the training corpus. I have managed to decrease the number of errors as showed in table 3, but didn't really designed the optimal set of templates that gave the expected result.

A closer look at the generated rules shows that many of the rules were not applied to an utterance as expected and many of the tags where not entirely accurate. For instance, template 9: *da:A>B* <- *u_mem:W@[0] & u_mem:W@[-1]*, that was pre considered to match the tag for the *clarify* move, resulted in the following rule: *da:reply_y>explain* <- *u_mem:as@[0] & u_mem:as@[-1].* The word "as" is repeated twice in the current utterance and the rule classifies this utterance as *explain*.

It seems, however, that the cooperation and exchange of ideas based on human knowledge provides many benefits for improving the performance of a dialogue act tagger.

Combining some features for dialogue act tagging I obtained a data accuracy of 64.5% and have shown that the design of templates for a simple dialogue act tagger can be achieved with the help of human knowledge. Despite the approach simplicity I have seceded to improve the performance of the tagger, which is encouraging.

These results give evidence that research in this direction is promising and worthwhile, one can further improve this understandable classification method by choosing a larger number of informants and by investigating different dialogue examples.

7. References

1. Brill, Eric (1995) *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging.* Computational Linguistics.
2. Lager, Torbjörn and Natalia Zinovjeva (1999) *Training a Dialogue Act Tagger with the μ-TBL System. Paper presented at the Third Swedish Symposium on Multimodal Communication*, Linköping University Natural Language Processing Laboratory (NLPLAB).
3. Samuel, Ken, Sandra Carberry and K. Vijay-Shanker (1998) *An Investigation of Transformation-Based Learning in Discourse.* In *Machine Learning: Proceedings of the 15th International Conference*.
4. Samuel, Ken, Sandra Carberry and K. Vijay-Shanker (1998) *Computing Dialogue Acts from Features with Transformation-Based Learning*. In *Applying Machine Learning to Discourse Processing,* AAAI Spring Symposium.
5. Samuel, Ken, Sandra Carberry and K. Vijay-Shanker (1998) *Dialogue Act Tagging with Transformation-Based Learning*. In *Proceedings of COLING/ACL'98*, pp. 1150-1156.
6. Samuel, Ken (1998) *Lazy Transformation-Based Learning*. In *Proceedings of the 11th International Florida Artificial Intelligence Research Symposium Conference*. University of Delaware, USA.
7. Anderson, Anne H., Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert (1991) *The HCRC Map Task Corpus*. University of Edinburgh and University of Glasgow, U.K.

8. Appendix

8.1. The twelve dialogue acts

Initiating moves
1– **Instruct**: S tells A to carry out an action
2– **Explain**: S states information not elicited by A
3– **Check**: S requests A to confirm information
4– **Align**: S checks for A’s attention or agreement
5– **Query-YN**: S asks A a yes/no-question
6– **Query-W**: A query that is not covered by the other move types

Response moves
7– **Acknowledge**: A verbal response that shows hearing, understanding or acceptance
8– **Reply-Y:** Any reply to a query, which means “yes” (expressed in any way)
9– **Reply-N**: Any reply to a query, which means “no” (expressed in any way)
10– **Reply-W**: Any reply to a query, that is not covered by the other move types
11– **Clarify**: A reply to some kind of question, containing information not strictly asked for
12- **Ready**: The closing move of a dialogue game. It also prepares for a new game

8.2. The fourteen templates

(1) da:A>B <- u_mem:W@[0].
(2) da:A>B <- u_first:W@[0]
(3) da:A>B <-u_last:W@[0].
(4) da:A>B <- u_bigram:W@[0].

(5) da:A>B <- da:C@[-1].
(6) da:A>B <- s:C@[0] & da:D@[-1].
(7) da:A>B <- s:C@[0] & u_mem:W@[0].

(8) da:A>B <- da:D@[-1,-2] & u_length:W@[0].
(9) da:A>B <- u_mem:W@[0] & u_mem:W@[-1].
(10) da:A>B <- u_first:W@[0]& u_length:C@[0].

(11) da:A>B <- u_mem:W@[0] & da:D@[-1].
(12) da:A>B <- s_change:C@[0] & da:D@[-1].

(13) da:A>B <- u_length:C@[0] & u_mem:W@[0].
(14) da:A>B <- u_length:C@[0] & da:D@[-1].