A Statistical Approach for Non-Sentential Utterance Resolution for Interactive QA System

Dinesh Raghu [*]
IBM Watson
diraghu1@in.ibm.com

Sathish Indurthi [*]
IBM Watson
saindurt@in.ibm.com

Jitendra Ajmera
IBM Watson
jajmera1@in.ibm.com

Sachindra Joshi
IBM Watson
jsachind@in.ibm.com

Abstract

Non-Sentential Utterances (NSUs) are short utterances that do not have the form of a full sentence but nevertheless convey a complete sentential meaning in the context of a conversation. NSUs are frequently used to ask follow up questions during interactions with question answer (QA) systems resulting into in-correct answers being presented to their users. Most of the current methods for resolving such NSUs have adopted rule or grammar based approach and have limited applicability.

In this paper, we present a data driven statistical method for resolving such NSUs. Our method is based on the observation that humans identify keyword appearing in an NSU and place them in the context of conversation to construct a meaningful sentence. We adapt the keyword to question (K2Q) framework to generate natural language questions using keywords appearing in an NSU and its context. The resulting questions are ranked using different scoring methods in a statistical framework. Our evaluation on a data-set collected using mTurk shows that the proposed method perform significantly better than the previous work that has largely been rule based.

1 Introduction

Recently Question Answering (QA) systems have been built with high accuracies [Ferrucci, 2012]. The obvious next step for them is to assist people by improving their experience in seeking day to day information needs like product support and troubleshooting. For QA systems to be effective and usable they need to evolve into conversational systems. One extra challenge that conversational systems throw is that users tend to form successive queries that allude to the entities and concepts made in the past utterances. Therefore, among other things, such systems need to be equipped with the ability to understand what are called Non-Sentential Utterances (NSUs) [Fernández et al., 2005, Fernández, 2006].

NSUs are utterances that do not have the form of a full sentence, according to the most traditional grammars, but nevertheless convey a complete sentential meaning. Consider for example, the conversation between a sales staff of a mobile store (S) and one of their customers (C), where C:2 and C:3 are examples of NSUs.

S:1 Hi, How may I help you

C:1 How much does an Apple iPhone 6 cost ?

S:2 $ . . .

C:2 What about 6S ?

S:3 $ . . .

C:3 with 64 GB ?

S:4 $ . . .

Humans have the ability to understand these NSUs in a conversation based on the context derived so far. The conversation context could include topic(s) under discussion, the past history between the participants or even their geographical location.

In the example above, the sales staff, based on her domain knowledge, knows that iPhone 6 and iPhone 6S are different models of iPhone and all phones have a cost feature associated with them. Therefore an utterance *What about 6S*, in the context of utterance *How much does an Apple iPhone 6 cost*, would mean *How much does an Apple iPhone 6S cost*. Similarly, 64 GB is an attribute of iPhone 6S and therefore the utterance *with 64*

[*] D. Raghu and S. Indurthi contributed equally to this work

Proceedings of the SIGDIAL 2015 Conference, pages 335–343,
Prague, Czech Republic, 2-4 September 2015.

GB in the context of utterance *How much does an Apple iPhone 6S cost* would mean *How much does an Apple iPhone 6s with 64 GB cost*.

In fact, studies have suggested that users of interactive systems prefer on being as terse as possible and thus give rise to NSUs frequently. Cognizant of this limitation, some systems explicitly ask the users to avoid usage of pronouns and incomplete sentences [Carbonell, 1983]. The current state of the QA systems would not be able to handle such NSUs and would result into inappropriate answers.

In this paper we propose a novel approach for handling such NSUs arising when users are trying to seek information using QA systems. Resolving NSUs is the process of recovering a full clausal meaningful question for an NSU utterance, by utilizing the context of previous utterances.

The occurrence and resolution of NSUs in a conversation have been studied in the literature and is an active area of research. However, most of the proposed approaches in the past have adopted a rule or grammar based approach [Carbonell, 1983, Fernández et al., 2005, Giuliani et al., 2014]. The design of the rules or grammars in these works were motivated by the frequent patterns observed empirically which may not scale well for unseen or domain specific scenarios.

Also, note that while the NSU resolution task can be quite broad in scope and cover many aspects including ellipsis [Giuliani et al., 2014], we limit the investigation in this paper to only the *Question* aspect of NSU, i.e. resolving C:2 and C:3 in the example above. More specifically, we would not be trying to resolve the system (S:2, S:3, S:4) and other non-question utterances (e.g. *OK, Ohh! I see*). This focus and choice is primarily driven by our motivation of facilitating a QA system.

We propose a statistical approach to NSU resolution which is not restricted by limited number of patterns. Our approach is motivated by the observation that humans try to identify the keywords appearing in the NSU and place them in the context to construct a complete sentential form. For constructing a meaningful and relevant sentence from keywords, we adapt the techniques proposed for generating questions from keywords, also known as keyword-to-question (K2Q).

The K2Q [Zhao et al., 2011, Zheng et al., 2011, Liu et al., 2012] is a recently investigated problem with the motivation to convert succinct web queries to natural language (NL) questions to direct users to cQA (community QA) websites. As an example, the query *ticket Broadway New York* could be converted to a NL question *Where do I buy tickets for the Broadway show in New York ?*. We leverage the core idea for the question generation module from these approaches.

The main contributions of this paper are as follows:

1. We propose a statistical approach for NSU resolution which is not limited by a set of predefined patterns. To the best of our knowledge, statistical approaches have not been investigated for the purpose of NSU resolution.

2. We also propose a formulation that uses syntactic, semantic and lexical evidences to identify the most likely clausal meaningful question from a given NSU.

In Section 2 we present the related work. We describe the a simple rule based approach in section 3. In section 4 we present the details of the proposed NSU resolution system. In Section 5, we report experimental results on dataset collected through mTurk and finally conclude our work and discuss future work in section 6.

2 Related Work

A taxonomy of different types of NSUs used in conversations was proposed by [Fernández et al., 2005]. According to their taxonomy the replies from the sales staff (S:2, S:3 and S:4) are NSUs of type *Short Answers*. However, the utterances C:2 and C:3 which are the focus of this paper and referred to as *Question NSU*, are not a good fit in any of the proposed types. One possible reason why the authors in [Fernández et al., 2005] did not consider them, may be because of the type of dialog transcripts used in the study. The taxonomy was constructed by performing a corpus study on the dialogue transcripts of the British National Corpus (BNC) [Burnard, 2000]. Most of the used transcripts were from meetings, seminars and interviews.

Some authors have also referred to this phenomenon as *Ellipsis* because of the elliptical form of the NSU [Carbonell, 1983, Fernández et al., 2004, Dalrymple et al., 1991, Nielsen, 2004, Giuliani et al., 2014]. While the statistical approaches have been investigated for the purpose of ellipsis detection [Fernández et al., 2004, Nielsen, 2004, Giuliani et al., 2014], it has been a common practice to use rules – syntactic or semantic – for the purpose of Ellipsis resolution [Carbonell, 1983, Dalrymple et al., 1991, Giuliani et al., 2014].

A special class of ellipsis, verb phrase ellipsis (VPE) was investigated in [Nielsen, 2004] in a domain independent manner. The authors have taken the approach of first finding the modal verb which can be then used as a substitute for the verb phrase. For example, in the utterance *"Bill loves his wife. John does too"*, the modal verb *does* can be replaced by the verb phrase *loves his wife* to result in the resolved utterance *"John loves his wife too"*. Authors used a number of syntactical features such as part-of-speech (POS) tags and auxiliary verbs, derived from the automatic parsed text to detect the ellipsis.

Another important class of NSUs referred to as *Sluice* was investigated in [Fernández et al., 2004]. Sluices are those situations where a follow-up bare *wh*-phrase exhibits a sentential meaning. For example:

Sue You were getting a real panic then.

Angela When?

Authors in [Fernández et al., 2004] extract a set of heuristic principles from a corpus-based sample and formulate them as probabilistic Horn clauses. The predicates of such clauses are used to create a set of domain independent features to annotate an input dataset, and run machine learning algorithms. Authors achieved a success rate of 90% in identifying sluices.

Most of the previous work, as discussed here, have used statistical approaches for detection of ellipsis. However, the task of resolving these incomplete utterances – NSU resolution – has been largely based on rules. For example, a semantic space was defined based on "CaseFrames" in [Carbonell, 1983]. The notion of these frames is similar to a SQL query where conditions or rules can be defined for different attributes and their values. In contrast to this, we present a statistical approach for NSU resolution in this paper with the motivation of scaling the coverage of the overall solution.

3 Rule Based Approach

As a baseline, we built a rule based approach similar to the one proposed in [Carbonell, 1983]. The rules capture frequent discourse patterns in which NSUs are used by users of a question answering system.

As a first step, let us consider the following conversation involving an NSU:

- **Utt1:** Who is the president of USA?
- **Ans1:** Barack Obama
- **Utt2:** and India?

We use the following two rules for NSU resolution.

Rule 1: if $\exists s | s \in phrase(Utt1) \wedge s.type = P_{Utt2}.type$ then create an utterance by substituting s with P_{Utt2} in the utterance $Utt1$.

Rule 2: if wh_{Utt2} is the only $wh-$word in $Utt2$ and $wh_{Utt2} \neq wh_{Utt1}$ then create an utterance by substituting wh_{Utt1} by wh_{Utt2} in $Utt1$.

Here $phrase(Utt1)$ denotes the set of all the phrases in $Utt1$ and P_{Utt2} denotes the key phrase that occurs in utterance $Utt2$. $s.type$ denotes the named entity type associated with the phrase s wh_{S1} and wh_{S2} denote the *wh* word used in the $Utt1$ and $Utt2$ respectively.

This rule based approach suffers from two main problems. One, it is only as good as the named entity recognizer (NER). For example, if *antonym ?* occurs in context of *What is the synonym of nebulous ?*, it is not likely for the NER to detect synonym and antonym are of the same type. Two, the approach has a very limited scope. For example, if *with 64 GB ?* occurs in context of *What is the cost of iPhone 6?*, the approach will fail as the resolution cannot be modeled with a simple substitution.

4 Proposed NSU Resolution Approach

In this section, we explain the proposed approach used to resolve NSUs. In the context of the running example above, the proposed approach should result in a resolved utterance *"Who is the president of India?"*. As mentioned above, intuitively the resolved utterance should contain all the keywords from *Utt2*, and these keywords should be placed in an appropriate structure created by the context of *Utt1*. One possible approach towards this would be to identify all the keywords from *Utt1* and *Utt2* and then forming a meaningful question using an appropriate subset of these keywords. Accordingly, the proposed approach consists of the following three steps as shown in Figure 1.

- Candidate Keyword Set Generation
- Keyword to Question Generation (*K2Q*)
- Learning to Rank Generated Questions

These three steps are explained in the following subsections.

4.1 Candidate Keyword Set Generation

Given *Utt1*, *Ans1* and *Utt2* as outlined in the previous section, the first step is to remove all the non-essential words (stop words) from these and generate different combinations of the essential words (keywords).

Let $U_2 = \{U_{2i}, i \in 1 \ldots N\}$ be the set of keywords in *Utt2* and $U_1 = \{U_{1i}, i \in 1 \ldots M\}$ be the set of keywords in *Utt1*. For the example above, U_2 would be $\{India\}$ and U_1 would be $\{president, USA\}$. Let Φ_{U_1,U_2} represent the power set resulting from the union of U_1 and U_2. Now, we use the following constraints to further rule out some invalid combinations:

- Filter out all the sets that do not contain all the keywords in U_2.
- Filter out all the sets that do not contain at least one keyword from U_1.

The basis for these constraints is coming from the observation that the NSU resolution is about interpreting the current utterance in the context of the conversation so far. Therefore it should contain all the keywords from the current utterance and at least one keyword from the context.

The valid keyword sets that satisfy these constraint are now used to form a meaningful question as explained in the following section.

4.2 Keyword to Question Generation

Keyword-to-question (K2Q) generation is the process of generating a meaningful and relevant question from a given set of keywords. For each keyword set $K \in \Phi_{U_1,U_2}$ resulting from the previous step, we use the following template based approach to generate a set of candidate questions.

4.2.1 Template Based Approach for K2Q

In this section, we summarize the template based approach proposed by [Zhao et al., 2011] that was adopted for this work. It consists of the following three steps:

- *Template Generation:* This step takes as input a corpus of reference questions. This corpus should contain a large number of example meaningful questions, relevant for the task or domain at hand. The keyword terms (all non-stop words) in each question are replaced by variable slots to induce templates. For example, questions *"what is the price of laptop?"* and *"what is the capital of India"* would induce a template *"what is the* T_1 *of* T_2*?"*. In the following discussion, we would denote these associated questions as Q_{ref}. Subsequently, the rare templates that occur less than a pre-defined threshold are filtered out.

  This step is performed once in an offline manner. The result of this step is a database of templates associated with a set of questions $\{Q_{ref}\}$ that induced them.

- *Template Selection:* Given a set of keywords K, this step selects templates that meet the following criteria:
  - The template has the same number of slots as the number of query keywords.
  - At least one question Q_{ref} associated with the template has one user keyword in exact same position.

  For example, given a query *"price phone"*, the template *"what is the T1 of T2"* would be selected, if there is a question *"what is the price of laptop"* associated with this template that has *price* keyword at the first position.

- *Question Generation:* For each of the templates selected in the previous step, a question Q is hypothesized by substituting the slot variables by the keywords in K. For example, if the keywords are *president, India* and the template is *"who is the T1 of T2"*, then the resulting question would be *" who is the president of India"*.

4.3 Learning to Rank Generated Questions

The previous step of question generation results in a set of questions $\{Q\}$ given a set of keywords $\{K\}$. To rank these questions, we transform each question's candidate into a feature vector. These features capture various semantic and syntactic aspects of the candidate question as well as the context. In this section we explain the different fea-

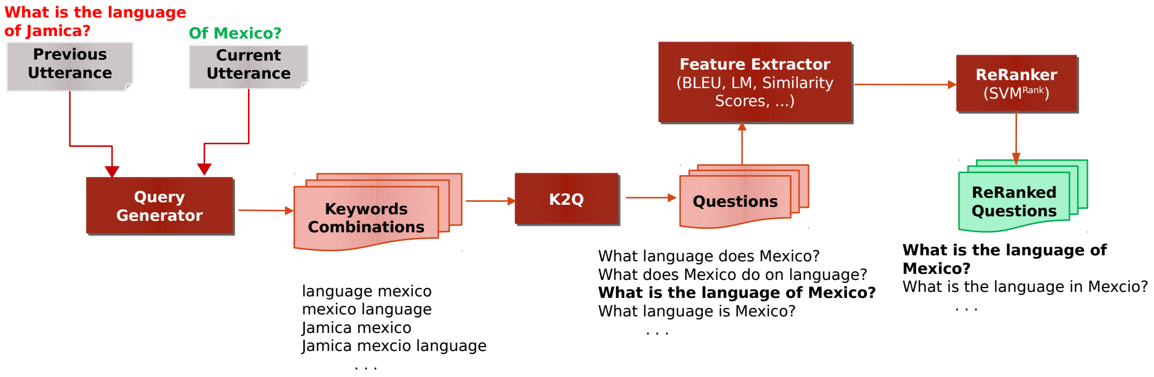
What is the language of Jamica?
Of Mexico?
Previous Utterance
Current Utterance
Query Generator
Keywords Combinations
K2Q
Questions
Feature Extractor (BLEU, LM, Similarity Scores, ...)
ReRanker (SVMRank)
ReRanked Questions
language mexico
mexico language
Jamica mexico
Jamica mexcio language
...
What language does Mexico?
What does Mexico do on language?
What is the language of Mexico?
What language is Mexico?
...
What is the language of Mexico?
What is the language in Mexcio?
...

Figure 1: Architecture of NSU Resolution System

tures and ranking algorithm used to rank the generated questions.

- **Semantic Similarity Score:** A semantic similarity score is computed between the keyword set K and each example question Q_{ref} associated with the template from which Q was generated. The computation is based on the semantic similarity of the keywords involved in Q and Q_{ref}.

$$Sim(Q, Q_{ref}) = \Pi_i^N Sim(K_i, Q_{ref,i})^{\frac{1}{N}} \tag{1}$$

where the similarity between the keywords involved $Sim(K_., Q_{ref,.})$ is computed as the cosine similarity of their word2vec representations [Mikolov et al., 2013].

- **Language Model Score:** To evaluate the syntactic correctness of the generated candidate question Q, we compute the language model score $LM(Q)$. A statistical language model assigns a probability to a sequence of n words (n-gram) by means of a probability distribution. The LM score represents how well a given sequence of n words is likely to be generated by this probability distribution. The distribution for the work presented in this paper is learned from the question corpus used in the template generation step above.

- **BLEU Score:** Intuitively, the intended sentential form of the resolved NSU should be similar to the preceding sentential form (*Utt1* in the example above). A similar requirement arises in evaluation of machine translation (MT) systems and BLEU score is the most commonly used metric for MT evaluation [Papineni et al., 2002]. We compute it as the amount of n-gram overlap between the generated question Q and the preceding utterance *Utt1*.

- **Rule Based Score:** Intuitively, the candidate question from K2Q should be similar to the resolved question generated by the rule based system (iff rules apply). As discussed in Section 3, we assign 1 to this feature when a rule fires, otherwise assign 0.

We use a learning to rank model for scoring each question $Q \in \{Q\}$, in the candidate pool for a given keyword set K: $w.\Psi(Q)$, where w is a model weight vector and $\Psi(Q)$ is the feature vector of question Q. The weights are trained using SVM^{rank} [Joachims, 2006] algorithm. To train it, for a given K, we assign higher rank to the correct candidate questions and all other candidates are ranked below.

5 Experiments

In this section, we present the datasets, evaluation approaches and results. We also present the comparative analysis of the performance obtained when we employ a rule-based baseline approach (Section 3) for this task.

5.1 Data

We organize the discussion around the data used for our evaluation in two parts. In the first part, we explain the dataset used for the purpose of setting up the template based K2Q approach described in Section 4.2. In the second part, we explain the dataset used for evaluating the performance of the NSU resolution.

Question	Answer	Q_{2e}	Q_{2r}
What does the golden marmoset eat?	flowers	and tiger?	What do tigers eat?
What is the average life span of Indian men?	65	And women	Average life span of women in India, is?
Who is the highest paid athlete today?	Tiger Woods	And in the 1990?	Who was the highest paid athlete in 1990?
Does a solid or liquid absorb more heat?	Liquid	What about gas or liquid?	Does a gas or a liquid absorb more heat?

Table 1: Examples of collected data entries from Amazon Mechanical Turk

5.1.1 Dataset for the K2Q Step

In section 4.2 we noted that the template generation step involves a large corpus of reference questions. One such large collection of open-domain questions is provided by the `WikiAnswers`* dataset.

The WikiAnswers corpus contains clusters of questions tagged by WikiAnswers users as paraphrases. Each cluster optionally contains an answer provided by WikiAnswers users. Since the scope of this work was limited to forming templates for the K2Q system, we use only the questions from this corpus. The corpus is split into 40 gzip-compressed files. The total compressed file size is 8GB. We use only the first two parts (out of 40) for the purpose of our experiments. After replacing the keywords by slot variables as required for template induction, this results into a total of $\approx 8M$ unique question-keyword-template tuples. Further, we filter out those templates which have less than five associated reference questions and this results into a total of $\approx 74K$ templates and corresponding $\approx 3.7M$ associated reference questions.

5.1.2 Dataset for NSU Resolution

In this section, we describe the data that we use for evaluating the performance of the proposed method for NSU resolution.

We used a subset of the data that was collected using Amazon Mechanical Turk. For collecting this data a question answer pair (Q,A) was presented to an mTurk worker and who was then asked to conceive another question Q_2 related to the pair (Q, A). The Q_2 was to be given in two different versions, an elliptical version Q_{2e} and a fully resolved version Q_{2r}. The original data contains 7400 such entries and contains examples for NSUs as well as anaphora in Q_2. We selected a subset of 500 entries from this dataset for our evaluation. Table 1 presents some examples entries from this data.

5.2 Evaluations

We present our evaluations based on the following three different configurations to investigate the importance of various scoring and ranking modules. The configurations used are,

1. **Rule Based:** This configuration is used as a baseline system, as described in section 3. As rule based methodologies are dominant in the field of NSU resolutions, we compare to clearly illustrate the limitations of just using rules.

2. **Semantic Similarity:** We investigate how well the semantic similarity score as described in Section 4.3 works when we sort the candidate questions generated based on this feature alone.

3. **SVM Rank:** In this configuration, we use all the scores as described in Section 4.3 in an SVM Rank formulation.

5.2.1 Evaluation Methodology

Given the input conversation $\{Utt1, Ans1, Utt2\}$, system generated resolved utterance Q (corresponding to NSU $Utt2$) and the intended utterance Q_r, the goal of the evaluation metric is to judge how similar Q is to Q_r. We use BLEU score and human judgments for the purpose of this evaluation.

BLEU score is often used for evaluation of machine translation systems to judge the goodness of the translated text with the reference text. Please note that we also used the BLEU score as one of the features as mentioned in Section 4.3. There, it was computed between the generated question Q and the preceding utterance $Utt1$. Whereas, for evaluation purposes, this score is computed between the generated question Q and the intended question provided by the ground truth Q_r.

To account for the paraphrasing errors, as the same utterance can be said in several different ways, we also use human judgment for the evaluation.

*Available at `http://knowitall.cs.washington.edu/oqa/data/wikianswers/`

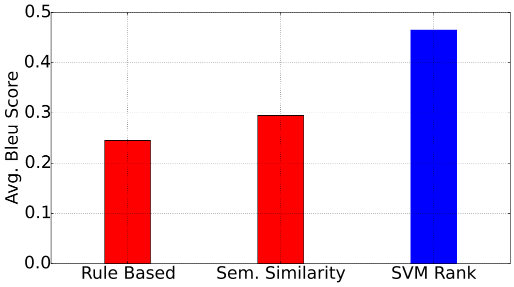

Figure 2: Average BLEU score for different configurations

Method	Recall@1
Rule Based	0.17
SVM Rank	**0.21**

Table 2: Comparing Recall@1 using Human Judgments

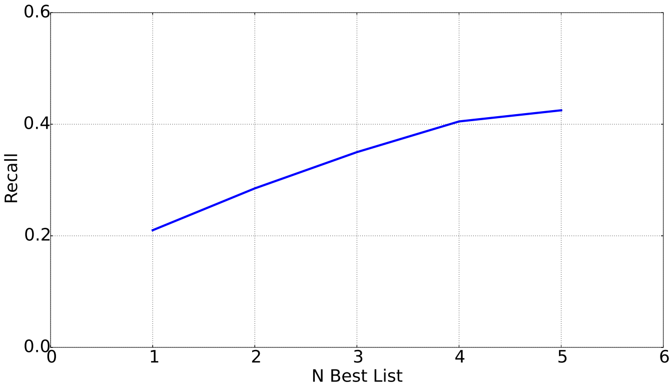

Figure 3: Recall@N obtained using human judgments

We use Recall@N to present the evaluation results when human judgments are used. Our test set comprises only of those utterances ($\{Utt2\}$) which require a resolution and therefore Recall@N captures how many of these NSUs were correctly resolved if candidates only up to top N are to be considered.

5.2.2 BLEU Score Evaluation

We compute the BLEU score between the candidate resolution Q and the ground truth utterance Q_r and compare it across the three configurations. Figure 2 shows the comparison of the average BLEU score at position 1. A low score for the rule based approach is expected as it resolves only those cases in which rules fire. The semantic similarity configuration gains over the rule based approach as it is able to utilize the template database generated using the WikiAnswers corpus. Finally, the SVM Rank uses various other scores (LM, BLEU score) on top of rule-based and semantic similarity score and therefore achieves higher BLEU Score.

5.2.3 Human Judgments Evaluation

Finally, to account for the paraphrasing artifacts manifested in human language, we use human judgments to make a true comparison between the rule based approach and the SVM Rank configuration.

For human judgments, we presented just the resolved Q and the ground truth Q_r. For all the 200 data points in the test set, top 5 candidates were presented to human annotators who were asked to decide if it was a correct resolution or not. We choose just the top 5 just to analyze the quality of the candidates generated at various positions by the system.

Table 2 shows the Recall@1 for the the two configurations. A better recall for the proposed

SVM configuration signifies the better coverage of the proposed approach beyond a pre-defined set of rules. The Recall@1 was used for this comparison since the rule-based approach can only yield a single candidate. To further see the behavior of the proposed approach as more candidates are considered, Recall@N is presented in Figure 3. The figure shows that a recall of 42.5% can be achieved when results up to top 5 are considered. The objective of this experiment is to study the quality of top (1-5) ranked generated questions. This experiment helps us conclude that improving the ranking module has the potential to improve the overall performance of the system.

5.3 Discussion

We discuss two types of scenarios where our SVM rank based approach works better than the baseline rule based approach. One of the rules to generate resolved utterance is to replace a phrase in *Utt1* with a phrase of the same semantic type in *Utt2*. Such an approach is limited by the availability of an exhaustive list of semantic types which is in general difficult to capture. In the following example, the phrases *antidote* and *symptoms* belong to the entity type *disease attribute*. However it may not be obvious to include *disease attribute* as a semantic type unless the context is specified. Our approach aims at capturing such semantic types automatically using the semantic similarity score.

Utt1 What is the antidote of streptokinase?

Utt2 What are the symptoms?

Resolved what are the symptoms of streptokinase

The baseline approach fails to handle cases where the resolved utterance cannot be generated by merely replacing a phrase in *Utt1* with a phrase in *Utt2*. While our approach can handle cases which requires sentence transformations such as the one shown below.

Utt1 Is cat scratch disease a viral or bacterial disease?

Utt2 What's the difference?

Resolved what's the difference between a viral and bacterial disease

One of the scenarios where our approach fails is when there are no keywords in *Utt2*. This is because the K2Q module tries to generate questions without any keywords (information) from *Utt2*. A few examples are given below.

Utt1 (a) Kansas sport teams?

Utt2 (a) What others?

Utt1 (b) Cell that forms in fertilization?

Utt2 (b) And ones that don't are called what?

6 Conclusion and Future Work

In this paper we presented a statistical approach for resolving questions appearing as nonsentential utterances (NSU) in an interactive question answering session. We adapted a keyword-to-question approach to generate a set of candidate questions and used various scoring methods to generate scores for the generated questions. We then used a learning to rank framework to select the best generated question. Our results show that the proposed approach has significantly better performance than a rule based method. The results also show that for many of the cases where the correct resolved question does not appear at the top, a correct candidate exists in the top 5 candidates. Thus it is possible that by employing more features and better ranking methods we can get further performance boost. We plan to explore this further and extend this method to cover other types of NSUs in our future work.

Acknowledgments

We thank Martin Schmid, IBM Watson Prague and Adam J Sporka, Pavel Slavik, Czech Techincal University Prague for providing us with the corpus of dialog ellipsis (dataset for NSU resolution) without which training and evaluation of our system would not have been possible.

References

Lou Burnard. Reference guide for the british national corpus. *Oxford University Computing Services*, 2000.

Jaime G. Carbonell. Discourse pragmatics and ellipsis resolution in task-oriented natural language interfaces. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, pages 164–168, 1983.

Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14:399–452, 1991.

Fernández, Raquel, Jonathan Ginzburg, and Shalom Lappin. Classifying ellipsis in dialogue: A machine learning approach. In *Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

Raquel Fernández. *Non-sentential utterances in dialogue: classification, resolution and use.* PhD thesis, University of London, 2006.

Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. Using machine learning for non-sentential utterance classification. pages 77–86, 2005.

David A Ferrucci. Introduction to this is watson. *IBM Journal of Research and Development*, 56 (3.4), 2012.

Manuel Giuliani, Thomas Marschall, and Amy Isard. Using ellipsis detection and word similarity for transformation of spoken language into grammatically valid sentences. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 243–250, 2014.

Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226, 2006.

Qiaoling Liu, Eugene Agichtein, Gideon Dror, Yoelle Maarek, and Idan Szpektor. When web search fails, searchers become askers: Understanding the transition. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 801–810, 2012.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations*, 2013.

Leif Arda Nielsen. Robust vpe detection using automatically parsed text. In *Proceedings of the ACL Workshop on Student Research*, 2004.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Shiqi Zhao, Haifeng Wang, Chao Li, Ting Liu, and Yi Guan. Automatically generating questions from queries for communitybased question answering. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 929–937, 2011.

Zhicheng Zheng, Xiance Si, Edward Y. Chang, and Xiaoyan Zhu. K2q: Generating natural language questions from keywords with user refinements. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 947–955, 2011.