SENTENCE SIMPLIFICATION FOR SPOKEN LANGUAGE UNDERSTANDING

Gokhan Tur Dilek Hakkani-Tür Larry Heck S. Parthasarathy

Speech at Microsoft | Microsoft Research
Mountain View, CA, 94041
gokhan.tur@ieee.org dilek@ieee.org {larry.heck,sarangp}@microsoft.com

ABSTRACT

In this paper, we present a sentence simplification method and demonstrate its use to improve intent determination and slot filling tasks in spoken language understanding (SLU) systems. This research is motivated by the observation that, while current statistical SLU models usually perform accurately for simple, well-formed sentences, error rates increase for more complex, longer, more natural or spontaneous utterances. Furthermore, users familiar with web search usually formulate their information requests as a keyword search query, suggesting that frameworks which can handle both forms of inputs is required. We propose a dependency parsing-based sentence simplification approach that extracts a set of keywords from natural language sentences and uses those in addition to entire utterances for completing SLU tasks. We evaluated this approach using the well-studied ATIS corpus with manual and automatic transcriptions and observed significant error reductions for both intent determination (30% relative) and slot filling (15% relative) tasks over the state-of-the-art performances.

Index Terms— spoken language understanding, intent determination, slot filling, semantic parsing, dependency parsing, sentence simplification

1. INTRODUCTION AND MOTIVATION

Spoken language understanding (SLU) in human/machine spoken dialog systems aims to automatically identify the intent of the user as expressed in natural language and extract associated arguments or slots [1] towards achieving a goal. An example utterance with semantic intent and slot annotations is shown in Table 1. The system can then decide on the next proper action to take according to the domain specific semantic template.

While these tasks can be seen as two halves of a whole, each of them have been studied in different contextual frameworks. Historically, intent determination has emerged from the call classification systems (such as the AT&T How May I Help You [2] system) after the success of the early commercial interactive voice response (IVR) applications used in call centers. On the other hand, the slot filling task originated mostly from non-commercial projects such as the DARPA (Defense Advanced Research Program Agency) sponsored Airline Travel Information System (ATIS) [3] project.

While both tasks have been extensively studied, it is still not possible to say that SLU is a solved problem, especially for more realistic, natural utterances spoken by a variety of speakers and for tasks more complex than simple flight information requests. Independent of the approach (data-driven vs. knowledge-based) employed for these tasks, the single biggest problem is the "naturalness" of the natural language input. This is apparent even in the artificially populated datasets such as ATIS.

Utterance	*How much is the cheapest flight arriving to JFK no later than tomorrow morning?*
Intent:	Airfare
Cost_Relative	*cheapest*
Destination_Airport	*JFK*
Arrive_Time.Relative	*no later than*
Arrive_Date.Relative	*tomorrow*
Arrive_Time.Period	*morning*

Table 1. An example utterance from the ATIS dataset.

Our previous work on error analysis for SLU using the ATIS corpus revealed that the most common reason of mistakes for these tasks along with their frequencies are mostly due to non-trivial syntactic characteristics [4]:

- *Intent Determination*:
  - *Prepositional phrases in noun phrases* (24.5%): These errors involve phrases where the prepositional phrase suggests a different intent than the actual one. For the example utterance "*Capacity of the flight from Boston to Orlando*", the actual intent is determined by the head word of the noun phrase, *capacity*, instead of *flight*.
  - *Wrong functional arguments of utterances* (30%): This category is similar to the previous one but the difference is that, instead of a prepositional phrase, the confused phrase is a semantic argument of the utterance. An example utterance would be "*What day of the week does the flight from Nashville to Tacoma fly on?*"
- *Slot Filling*:
  - *Long distance dependencies* (26.9%): These are slots where the disambiguating tokens are out of the current n-gram context. For example, in the utterance "*Find flights to New York arriving in no later than next Saturday*", a 6-gram context is required to resolve that *Saturday* is the arrival date.

These error categories for SLU have previously been addressed in the literature, and can be clustered into two groups, depending on whether or not a syntactic parser is used.

Raymond and Riccardi [5] extracted features using manually-designed patterns. For example, they used the existence of the verb "arrive" in the sentence and framed this as using *a priori* knowledge for SLU. This improved the slot filling performance from 95.0% to 95.6% for ATIS.

Similarly, Jeong and Lee [6] used trigger patterns. The slot filling performance increased from 94.8% to 96.2% for the Communicator corpus. They also tried to exploit syntactic information, such as the head word, without success.

Moschitti et al. [7] presented the first study showing the use of using syntactic features for slot filling via syntactic tree kernels with support vector machines. This improved the performance in ATIS to 95.9% from 95.5%.

Regarding intent determination, in our previous work we have presented an approach populating heterogeneous features from syntactic and semantic graphs of utterances [8]. We showed their use in a cascaded setup for utterances which received low confidences using the baseline word n-gram based classifier.

In this discussion we should not omit knowledge-based semantic parsing approaches, such as SRI Gemini [9], MIT TINA [10] and CMU Phoenix [11] systems which heavily rely on syntactic parsing.

As seen, while using syntactic information for SLU is not a novel idea, in this paper we propose going one step further and propose using syntactic information to modify the input for SLU tasks. There are two main reasons that motivate us for sentence simplification for SLU:

- First, as observed by [12], the major problem with using syntactic features for classification is that the paths in the parse trees occur relatively infrequently (or not at all) in the training set. A simple negation, for example, may totally change the structure of the syntactic parse tree. Sentence simplification may then help this problem by condensing the training and test sets so that the classifier will work better as the average frequency of candidate lexical and syntactic features increase.
- Second, this will alleviate the problem of handling long-distance dependencies. Given that most classifiers rely on word n-grams, where n is typically less than 5 words, it is critical to cover such cases without bombarding the classifier with candidate syntactic features.

Our approach can be seen as an utterance compression task, where the goal is to rephrase the same intent with fewer words, and by doing so, would also support short, keyword sequence inputs. This is analogous to understanding keyword-based queries where there is usually a natural language query in mind. An example would be rephrasing the query "*What is the capacity of a 737?*" as "*capacity 737*". While sentence simplification intuitively makes more sense for intent determination, which is typically framed as an utterance classification task, we also demonstrate that this approach is effective for slot filling due to its power for handling long distance dependencies. For both tasks, our approach relies on features extracted from the dependency parse of the input utterance.

In the next section, we will present the sentence simplification approach we used in this study. Then we describe how this is used for intent determination and slot filling in sections 2.1 and 2.2, respectively. We present our experimental results in Section 3 before concluding in Section 4.

2. SENTENCE SIMPLIFICATION

Sentence simplification is an area which has been studied for various language processing tasks such as summarization or semantic role labeling, with different motivations. For summarization, the motivation is presenting the information to the user in as few words as possible. For example, [13] employed sentence simplification to get rid of certain patterns such as noun appositives, nonrestrictive relative clauses, intra-sentential attributions, or lead adverbials and conjunctions from newspaper articles.

Our approach is actually more similar to [12], who simplifies sentences for better classification (in their case better semantic role labeling) via a number of hand-written syntactic parse tree transformations. For example, one transformation converts passive sentences into active voice, another eliminates negations, and so on. During run-time they combine the outputs of the models trained from original and simplified sentences. Note that, in our case the simplified sentence does not need to be grammatical as the consumer is not a human but a classifier. This gives greater flexibility to assess possible simplification strategies and enables us to perform quantitative experiments in an efficient manner.

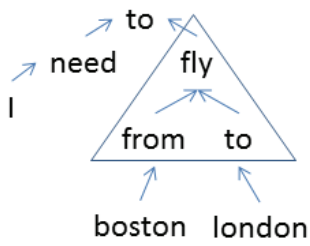

Fig. 1. Dependency parse of an example sentence "*I need to fly from Boston to London*" and demonstration of simplification.

The simplification procedure relies on dependency parses of the sentences where the structure of a sentence is determined by the relation between a word (a head) and its dependents. Each word has a head it is pointing to. For example, for the noun phrase *blue book*, *blue* points to *book*.

In this study we employ the Berkeley Parser [14], a state-of-the-art parser trained from a treebank following a latent variable approach by iteratively splitting non-terminals to better represent the data. We use the LTH Constituency-to-Dependency Conversion toolkit[1] to form dependency parses from the output parse trees. To adapt the parser to the speech domain, we retrain it using monocase WSJ treebank stripping out punctuation [15] and further employ a self-training approach using the ATIS training data. This process improves the parser's ability to handle monocase words, lack of punctuation and focus on conversational style sentences which rarely occur in textual corpora.

2.1. Intent Determination

The key observation is that, for intent determination, the presence or absence of some salient phrases is critical for making the correct classification. In the marginal case, if each intent was uttered by just its name, there would be no classification mistakes. Since this is not possible, the approach we take is approximating this behavior via sentence simplification based on syntactic information using a dependency parser as explained above.

Given the dependency parse of a sentence, the simplification algorithm then only uses the top level predicate and its dependents, excluding the auxiliary predicates, such as *need* or *want*. Consider the example sentence "*I need to fly from London to Boston*". Its dependency parse is shown in Figure 1. The highest level non-auxiliary predicate is *fly*. Its dependents are *from* and *to*. Then this sentence is simplified as *fly from to*. The prepositions are kept as they are salient for many intents. The power of using simplification comes from the fact that, it gets rid of phrases which complicates the classification task. If the example sentence is "*how much does it cost to fly from*

[1] http://nlp.cs.lth.se/software/treebank_converter/

London to Boston", the word *fly* is no longer the highest level predicate and the sentence is simplified as *cost to*. Now, this is a much simpler job for the intent classifier.

For intent determination, we use icsiboost[2], an implementation of the AdaBoost.MH algorithm, a member of the boosting family of classifiers [16]. Boosting is an iterative procedure that builds a new weak learner h_t at each iteration. Every example of the training data set is assigned a weight. These weights are initialized uniformly and updated on each iteration so that the algorithm focuses on the examples that were wrongly classified during the previous iteration. At the end of the learning process, the weak learners used on each iteration t are linearly combined to form the classification function:

$$f(x,l) = \sum_{t=1}^{T} \alpha_t h_t(x,l)$$

with α_t the weight of the weak learner h_t and T the number of iterations of the algorithm.

2.2. Slot Filling

For slot filling, our main motivation is handling the long distance dependencies between the slot phrase and its disambiguator. A disambiguator is a phrase which determines the semantic subcategory of an entity. For the example in Table 1, the word *morning* is known to be a time period. But the semantic disambiguation of whether it is an arrival or departure time relies on the predicate of the sentential clause, i.e., *arriving*. This information is straightforward to capture using the dependency parse tree of this sentence but it may get lost among other features when constituency parse trees are used as tree kernel features as in Moschitti et al. [7].

To this end, for slot filling, predicates of sentential clauses in addition to top level non-auxiliary predicates are considered. In the example sentence "*Find flights departing from New York tomorrow arriving in Tokyo no later than Saturday*", the predicate *arrive* is also considered as a feature while classifying the words which directly or indirectly depend on it, in this case *in Tokyo no later than Saturday*. The same is true for the phrase *from New York tomorrow* for the predicate *departing*. The recursive algorithm to find the predicate head of a given word is as follows: If the head of a word is a predicate, then it is used, otherwise, the predicate head of its head is used as its predicate head.

For slot filling, following [5], the baseline statistical model relies on word n-gram based linear chain conditional random fields (CRF). CRFs are shown to outperform many other classification methods for sequence classification since the training can be done discriminatively over a sequence. Similar to maximum entropy models, the conditional probability, $p(Y|X)$ is defined as:

$$p(Y|X) = \frac{1}{Z_X} exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(y_{t-1}, y_t, x, t) \right)$$

with the difference that both $X = x_1, ..., x_T$ and $Y = y_1, ..., y_T$ are sequences instead of individual local decision points given a set of features f_k with associated weights λ_k. Z_X is the normalization term. We used the CRF++ toolkit[3] in this study.

[2]http://code.google.com/p/icsiboost/

[3]http://crfpp.sourceforge.net

Approach	*TCER (Man)*	*TCER (ASR)*
1. Baseline (Word n-grams)	4.25%	7.95%
2. Heads only	16.68%	22.17%
1+2	4.81%	6.94%
3. Head/Dependency pairs only	10.19%	12.76%
1+3	4.47%	7.05%
4. Simplified sentence only	7.27%	8.28%
1+4	3.24%	5.59%
4 with retrained parser	6.60%	8.84%
1+4 with retrained parser	**3.02%**	**5.37%**

Table 2. Intent determination experiments on the ATIS set using both manual transcriptions (man) and speech recognition outputs (ASR).

3. EXPERIMENTS AND RESULTS

In this paper, we use the ATIS corpus as used in He and Young [17] and Raymond and Riccardi [5]. The training set contains 4,978 utterances selected from the Class A (context independent) training data in the ATIS-2 and ATIS-3 corpora, while the test set contains 893 utterances from the ATIS-3 Nov93 and Dec94 datasets. Each utterance has its named entities marked via table lookup, including domain specific entities such as city, airline, airport names, and dates.

The corpus has 17 different intents, such as *Flight* or *Aircraft capacity*. The prior distribution is, however, heavily skewed, and the most frequent intent, *Flight* represents about 70% of the traffic. The ATIS utterances are represented using semantic frames, where each sentence has a goal or goals (a.k.a. intent) and slots filled with phrases. The values of the slots are not normalized or interpreted. In total there are 2,837 slots belonging to 69 different categories to fill.

The ATIS corpus is automatically recognized using the generic dictation models using the Microsoft commercial speech recognition system. The word error rate was 13.76% without using the ATIS training set. While this is significantly higher than the best reported performances of about 5% WER [18], this provides a more challenging and realistic framework for syntax driven studies.

3.1. Intent Determination

Table 2 shows the results for intent determination using the sentence simplification technique. For evalution, the error rate of the top scoring class (TCER) is used. The baseline performance of 4.25% is obtained using only word trigrams with Boosting and is already significantly better than the previously published TCER of 4.81% by [18] using Maximum Entropy classifier on the same dataset. One important result is that, using simplified sentences alone did not improve the performance, but when it is combined with the actual sentence, the TCER significantly[4] reduced to 3.02% (about 30% relative). When ASR outputs are used, even higher relative reductions in TCER is observed, proving the robustness of the parser and the approach used. We also present results using all and top level head/dependency pairs as features and combine them with word n-grams. These results show that when used in a straightforward fashion, head/dependency pairs do not help the intent classification performance, and actually hurts performance since this introduces more confusion to the model.

In the ATIS test set, only a few sentences have been correctly classified with the baseline model and erroneously classified by the model trained also with simplified sentences.

[4]According to the Z-test with 0.95 confidence interval.

Class	*Original (%)*	*Simplified (%)*
Abbreviation	2/32 (6.25%)	2/32 (6.25%)
Aircraft	1/8 (1.25%)	0/8 (0.00%)
Airfare	3/52 (5.76%)	5/52 (9.61%)
Airline	1/38 (2.63%)	0/38 (0.00%)
Airport	2/18 (1.11%)	0/18 (0.00%)
Capacity	9/21 (42.85%)	4/21 (19.04%)
City	3/5 (60.00%)	2/5 (40.00%)
Day Name	2/2 (100.00%)	2/2 (100.00%)
Distance	1/10 (10.00%)	1/10 (10.00%)
Flight	16/640 (2.50%)	6/640 (0.94%)
Flight No	4/9 (44.44%)	3/9 (33.33%)
Flight Time	1/1 (100.00%)	0/1 (0.00%)
Ground Fare	2/7 (28.57%)	2/7 (28.57%)
Ground Service	0/36 (0.00%)	0/36 (0.00%)
Meal	5/5 (100.00%)	4/5 (80.00%)
Quantity	0/8 (0.00%)	0/8 (0.00%)
Restriction	0/1 (0.00%)	0/1 (0.00%)
Total	38/893 (4.25%)	27/893 (**3.02%**)

Table 3. Class-level intent classification error rates on the ATIS set using manual transcriptions.

Table 3 shows the class-level classification error rates. We see that the greatest reductions in error rates are for the *Capacity* and *Flight* classes, which contribute to the overall error rate reduction.

3.2. Slot Filling

Table 4 presents the effectiveness of the sentence simplification for the slot filling task. The data sets are converted into the IOB format so that there is only one word per sample to classify. Using the CoNLL evaluation script[5], the F-Measure we obtained is 94.4% using all trigrams[6], which is comparable to what has been reported in the literature (e.g., [5]).

This table shows the importance of using the predicate head instead of the immediate head for slot filling. Similar to [6], we observed little improvement using immediate head with manual transcriptions which disappeared using ASR output.

The use of dependency parse information significantly[7] decreases the token error rate (TER) by about 15% relative from 2.23% to 1.91% for manual transcriptions. This corresponds to an increase of 0.6% absolute for the F-Measure. Using ASR output, F-Measure increases 0.7% absolute.

When we look at slot-level performances, we see that the biggest improvement comes from the disambiguation of *depart* and *arrive* types of the slots. More specifically, the macro-average of F-Measures for depart/arrive related 21 slots increase from 70.0% to 84.5%. These slots account for about 26% of the total.

4. CONCLUSIONS

We present a dependency parsing based sentence simplification method and demonstrated its use to improve intent determination and slot filling tasks in spoken language understanding (SLU) systems. The simplicity and effectiveness of this approach motivates us to pursue it further. One idea is incorporating semantic role labeling for semantically motivated simplification. Another idea is running slot filling first and using the output for better simplification for intent determination. Note that this method can be easily applied to other language processing tasks such as dialog act tagging, summarization, or machine translation.

[5]http://www.cnts.ua.ac.be/conll2000/chunking/output.html

[6]It is 94.9% using the representation used by [5], who reported 95.0%

[7]According to the McNemar significance test [19], $p < 0.001$

Approach	*F-M* (man)	*TER* (man)	*F-M* (ASR)	*TER* (ASR)
Baseline (Word n-grams)	94.4%	2.23%	88.9%	4.25%
+Immediate Head words	94.7%	2.14%	88.9%	4.28%
+Predicate Head words	**95.0%**	**1.91%**	**89.6%**	**3.93%**

Table 4. Slot filling experiments on the ATIS set using manual transcriptions (man) and speech recognizer output (ASR).

5. REFERENCES

[1] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding - an introduction to the statistical framework," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, September 2005.

[2] A. L. Gorin, G. Riccardi, and J. H. Wright, "How May I Help You?," *Speech Communication*, vol. 23, pp. 113–127, 1997.

[3] P. J. Price, "Evaluation of spoken language systems: The ATIS domain," in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Hidden Valley, PA, June 1990.

[4] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?," in *Proceedings of the IEEE SLT Workshop*, Berkeley, CA, 2010.

[5] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proceedings of the Interspeech*, Antwerp, Belgium, 2007.

[6] M. Jeong and G. G. Lee, "Exploiting non-local features for spoken language understanding," in *Proceedings of the ACL/COLING*, Sydney, Australia, July 2006.

[7] A. Moschitti, G. Riccardi, and C. Raymond, "Spoken language understanding with kernels for syntactic/semantic structures," in *Proceedings of the IEEE ASRU Workshop*, Koyoto, Japan, 2007.

[8] D. Hakkani-Tür, G. Tur, and A. Chotimongkol, "Using syntactic and semantic graphs for call classification," in *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, Ann Arbor, MI, June 2005.

[9] J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran, "Gemini: A natural language system for spoken language understanding," in *Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, NJ, March 1993.

[10] S. Seneff, "TINA: A natural language system for spoken language applications," *Computational Linguistics*, vol. 18, no. 1, pp. 61–86, 1992.

[11] W. Ward and S.Issar, "Recent improvements in the CMU spoken language understanding system," in *Proceedings of the ARPA HLT Workshop*, March 1994, pp. 213–216.

[12] D. Vickrey and D. Koller, "Sentence simplification for semantic role labeling," in *Proceedings of the ACL*, 2008.

[13] L. Vanderwende, H. Suzuki, and C. Brockett, "Microsoft research at DUC2006: task-focused summarization with sentence simplification and lexical expansion," in *In Proceedings of the DUC*, 2006.

[14] S. Petrov and D. Klein, "Learning and inference for hierarchically split PCFGs," in *Proceedings of the AAAI*, 2007.

[15] B. Favre, D. Hakkani-Tür, S. Petrov, and D. Klein, "Efficient sentence segmentation using syntactic features," in *Proceedings of the IEEE SLT Workshop*, Goa, India, 2008.

[16] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[17] Y. He and S. Young, "A data-driven spoken language understanding system," in *Proceedings of the IEEE ASRU Workshop*, U.S. Virgin Islands, December 2003, pp. 583–588.

[18] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, and A. Acero, "An integrative and discriminative technique for spoken utterance classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1207–1214, 2008.

[19] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proceedings of the ICASSP*, Glasgow, Scotland, 1989.