COUPLING SPEECH RECOGNITION AND RULE-BASED MACHINE TRANSLATION WITH CHART PARSING

Selçuk Köprü, Adnan Yazıcı**, Tolga Çiloğlu***, Ayşenur Birtürk***

*Applications Technology, Inc., Turkey
**Dept. of Computer Engineering, Middle East Technical University, Turkey
***Dept. of Electrical and Electronics Engineering, Middle East Technical University, Turkey

ABSTRACT

This article presents our approach and findings in coupling statistical Speech Recognition (SR) systems with a rule-based Machine Translation (MT) system. Most of the literature about coupling focuses on how to integrate SR with statistical MT systems. We think that utilizing rule-based MT systems for Speech Translation (ST) task is important and still remains as an open research issue. In this paper we introduce the Apptek Speech Translator system, the distinctive approach used for coupling SR and rule-based MT, and the results of the experiments to justify the approach.

Index Terms— chart parsing, machine translation, coupling, speech translation, speech recognition

1. INTRODUCTION

The demand for integrating SR systems and MT systems is becoming more and more important as both technologies are advancing towards satisfactory levels. Realizing ST task requires this integration to be as much perfect as possible. Most of the research focuses on how to integrate SR with statistical MT systems. The fact that both are based on the same principles and techniques might be a reason for existing integrations. This allows both systems to interact and exchange information in a much more straightforward way. Quite few of the studies [6, 12, and 13] address coupling heterogeneous components of NLP Systems. In this respect, our research work is unique because it presents a new approach and findings in coupling statistical SR systems with a rule-based MT system. It would be very beneficial to utilize rule-based MT systems for ST task because linguistic resources are used extensively. The rules and lexicons are created in many years based on broad studies and experience. Thus, utilizing these existing linguistic resources for both coupling and analysis would be a big gain in ST.

There is a wide range of studies available in literature attacking the coupling problem. Reference [10] is the first study classifying coupling into three categories: tightly-coupled, loosely-coupled, and semi-coupled. According to [8], tightness describes how close the SR and MT units interact with each other. In a tightly-coupled system, speech and MT processing is packed into an inseparable unit. In a loosely-coupled system, processing is inside independent modules. Finally, semi-coupled systems lie between the previous two approaches in terms of isolation. Tight coupling is possible if both modules are statistically based because the unit of information interchanged between systems is meaningful for both sides. Thus, it is considered to be a difficult task to tightly couple a statistical SR with a rule-based MT.

The coupling method suggested in [2] is a tightly coupled system where the whole process is based on Bayes decision rule. The work in [3] and [4] are similar to [2] and all are applicable only for statistical MT systems. In [5], Saleem et. al. discuss another approach towards tightly coupling SR and statistical MT systems. They conclude that using word graphs as the information exchange unit does improve performance when the weighted acoustic scores are incorporated into the MT unit. An alternative for a word-graph is a confusion network which is another type of directed graph where each path from start to end includes all existing nodes. Using confusion networks as the unit of exchanged information between SR and statistical MT is explored in [7] and [9].

While integrating the SR system with the rule-based MT system, this study uses word-graphs and chart parsing with new extensions to achieve the coupling. Parsing of word lattices has been a topic of research over the past decade [1, 11, 13, 14, and 15]. The idea of chart parsing the word graph in SR systems has been used previously in different studies in order to resolve ambiguity [1, 11]. However, to the best of our knowledge, the specific method for chart parsing a word graph introduced in this paper has not been used for coupling purposes before. There are two main differences between the work presented here and the previous ones existing in literature. First, in [1] and [11], the chart is populated with the same word-graph that comes from the speech recognizer without any pruning, whereas in our approach the word-graph is shrunken to an acceptable size. Otherwise, the efficiency becomes a big challenge because the search space introduced by a chart with more than thousand initial edges can be easily beyond current practical limits. Another important difference in our approach is the extension of the chart to eliminate spurious parses. Main distinction between [13, 14, and 15] and our study is the parsing algorithm being used. In contrast to our chart parsing approach augmented by unification based feature structures, Charniak parser is used in those studies along with PCFG.

In the next section we introduce the Apptek Speech Translator and our approach on coupling. In Section 3, we present the results of the experiments that are carried out to justify the approach. The concluding remarks are given in Section 4.

2. APPTEK SPEECH TRANSLATOR

The general architecture of the Apptek Speech Translation system is depicted in Figure 1. The system is loosely coupled, i.e., there is a one directional information flow between the SR and MT. The original word-graph created by the SR is reduced according to the Viterbi search algorithm based on bigram scores inside the pruning module. The output is another word-graph representing the N-best sentence hypotheses. The pruned word graph is processed by the MT component of the speech translation system. MT task deploys a transfer based approach and processing is divided into three clear-cut phases: analysis, transfer and generation. At the end of the analysis, any ambiguity is resolved and the best sentence hypothesis is picked for the transfer stage. In this paper, our focus is on the part encircled with dashed lines in Figure 1.

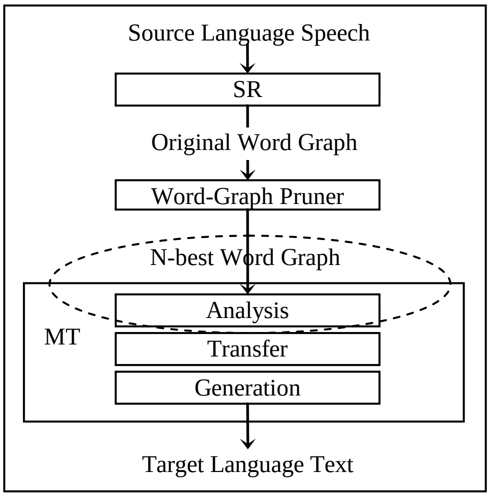

Figure 1. Apptek Speech Translator.

2.1. MT Analyzer

The analysis is accomplished in two consecutive tasks. First, morphological analysis is performed on the word level and any information carried by the word is extracted to be used in later stages. Next, syntactic analysis is performed on the sentence level. The syntactic analyzer consists of a chart parser in which the rules modeling the source language grammar are augmented with feature structures. The grammars are implemented using *Lexical Functional Grammar* (LFG) paradigm. Primary data structure to represent the features and values is a directed acyclic graph (dag). The system also includes an expressive Boolean formalism, used to represent functional equations to access, inspect or modify features or feature sets in the dag. Complex feature structures, e.g. lists, sets, strings, and conglomerate lists, can be associated with lexical entries and grammatical categories using inheritance operations. Unification is used as the fundamental mechanism to integrate information from lexical entries into larger grammatical constituents.

A sample parse tree and the feature structures in English are shown in Figure 2. For the case of simplicity, many details and feature values are not given. The dag containing the information originated from the lexicon and the information extracted from morphological analysis is shown on the leaf levels of the parse tree in Figure 2. The final dag corresponding to the root node is built during the parsing process in cascaded unification operations specified in the grammar rules.

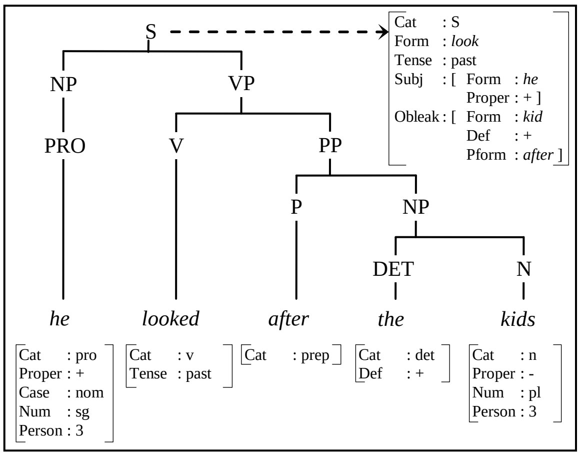

Figure 2. Unification based parsing.

2.2. Approach

In a loosely coupled system, information flow between modules can be made in different forms and quantities. In the simplest form, the SR provides only the first-best sentence. It can also provide an N-best list of sentence hypotheses to the parser to eliminate existing recognition errors [16]. In a more sophisticated form, the SR provides a word graph containing more information compared to the previous two forms.

The fundamental idea behind our solution is to initialize the chart of the MT parser using the simplified word graph coming from the Word-Graph Pruner module. Thus, all selected sentence hypotheses will be processed simultaneously. Initialized chart will be processed once until the first sentence hypothesis is picked by the parser. In its basic form, the chart models a confusion network, which might lead to spurious parse trees. We have extended the original chart representation and its processing in order to be able to model a word-graph instead. The advantage of the approach is essentially in the ability to rule out non-syntactic hypotheses in a parallel fashion.

The chart initialization algorithm assumes that a word-graph is represented by (N, S, A) where N is a set of nodes, S is a subset of N and contains the starting nodes, and A is a list of arcs as in $u \rightarrow v$, where $u, v \in N$. Algorithm 1 ensures the creation of a valid chart that can be processed by the well-known chart parser. The number of rows in the chart is equal to the number of sentence hypotheses in the input word-graph. The algorithm makes use of a stack to keep track of visited nodes and the associated cell indices. *pop* and *push* functions insert into and retrieve data from the stack data structure. The chart is represented by a table like structure. Function *split_cell(x, y, i)*, splits all the cells from position *(x, y)* until *(x, z)* into *i* vertical cells, where *z* is the total number of columns in the table. Splitting operation is a way of inserting new rows to the table. While doing this operation, all effected row numbers inside the stack are updated. Function *merge_cells (x, y, z)*, merges cells in the range from *(x, y)* to *(x, z)*. Function *out_arcs(n)*, gives the number of arcs in A where n is the source node. Finally, function *column(n)* returns the column index of node n in the chart.

Algorithm 1. Initialization of the chart

```
for each node n Є S
    row ← 1
    push (n, row, 1)
    row ← row + 1
while stack ≠ Empty
    pop (n, row, col)
    chart [row, col] ← n
    if out_arcs(n) > 1
        split_cell (row, col + 1, out_arcs(n))
    i ← 0
    for each arc: n → m
        if all nodes p exist in chart, where ∃ an arc: p → m
            push (m, row + i, col + 1)
        i ← i + 1
    for each arc: p → n
        if column (p) + 1 ≠ col
            merge_cells (row, column (p), col - 1)
```

Consider the simple word-graph, the corresponding chart and the hypotheses depicted in Figure 3. Using Algorithm 1, the chart is populated for syntactic analysis. The *for* loop at the beginning of the Algorithm 1 puts starting nodes into different rows. The *while* loop processes the remainder of the path to the final node. Words on the same column are regarded as a single lexical entry with different senses (e.g. *'boy'* and *'boycott'* at column 2). Words spanning more than one column are regarded as idiomatic entries (e.g. *'escalated'* at columns 3 to 5). Merged cells in the chart (e.g. *'the'* and *'yesterday'* at columns 1 and 6, respectively) are shared in the two sentence hypotheses.

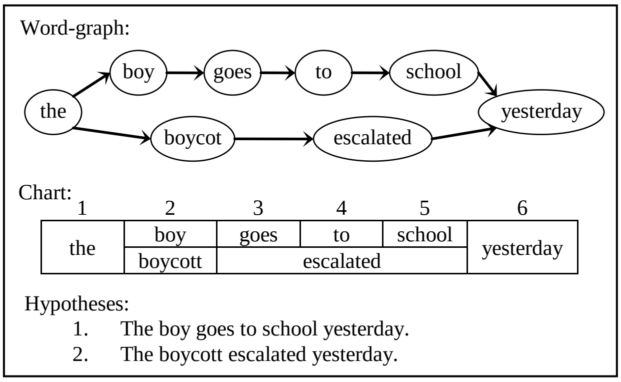

Figure 3. Sample word-graph, the corresponding chart and the hypotheses.

In an ordinary chart parser, the initial chart in Figure 3. can produce some spurious parses. For example, both of the entries at location 2,*'boy'* and *'boycott'*, can be combined with the word *'goes'*, although *'boycott goes'* is not allowed in the original word-graph. We have eliminated these kinds of spurious parses by introducing a new feature *rowid* into the lexical dags and grammar rules. This feature will contain the row number of the words. Constituents can be combined only if their *rowid* values can be unified. *rowid* for a cell spanning more than one row will include all the spanned row numbers as a set. The sample implementation of this idea is illustrated in Figure 4. *'boycott'* and *'goes'* cannot be combined in a parse tree because their *rowid* values do not unify. *'the'* can be combined with both *'boy'* and *'boycott'* because its *rowid* value contains 1 and 2. This extension to the parser makes our approach word-graph based rather than confusion network based.

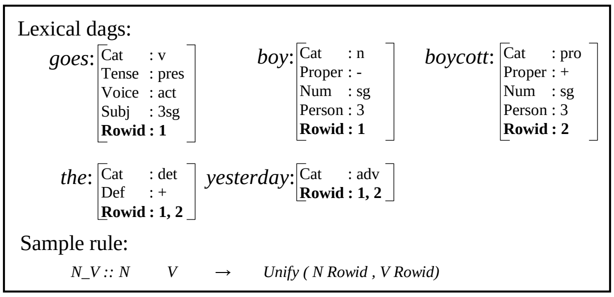

Figure 4. Sample rule enforces *rowid* labels from the constituents to be unified.

3. RESULTS OF EXPERIMENTS

In this section we present some experiments carried out on English to Arabic translation as a part of the Apptek speech translation project. The English and Arabic monolingual lexicons contain 40K and 60K entries, respectively. English to Arabic bilingual transfer lexicon contains also 40K entries. English grammar is modeled with 125 parsing rules implemented in LFG formalism.

Table 1. represents the number of complete and incomplete edges generated for the sample word-graph in Figure 3 for all three approaches. "Gain" column in the table represents the decrease in the total number of edges compared to the N-best list approach. The first-best approach performs better only if the first hypothesis is syntactically correct (first row in Table 1). When the first hypothesis is syntactically ungrammatical in the first-best approach (second row in Table 1), the N-best word-graph processing system performs better and produces more accurate output. The reason for this is because it is able to parse the first N sentence hypotheses together instead of trying to parse an ungrammatical sentence.

Table 1. Number of edges generated in the two different approaches for the sample in Figure 3.

Approach	Complete Edges	Incomplete Edges	Total	Gain
First-best (*success*)	62	236	298	75%
First-best (*fail*)	92	849	941	19%
N-best List	150	1007	1157	-
N-best Word-Graph	78	284	362	69%

Compared to the N-best list approach, the shared edges are processed only once for all hypotheses. This saves a lot on the number of complete and incomplete edges generated during parsing. Hence, the overall processing time required to analyze the hypotheses are reduced. In an N-best list approach, where each hypothesis is processed separately in the analyzer, there are different charts and different parsing instances for each sentence hypothesis. Shared words in different sentences are parsed repeatedly and same edges will be created at each instance. As it can be seen, there is a 69% gain in the number of edges if the N-best word-graph approach is used instead of the N-best list approach for the sample in Figure 3. The huge difference in the number of incomplete edges arises from the fact that the processing continues until a successful parse is found. For a syntactically wrong sentence, the search for the successful tree continues until all possibilities are exhausted. The word-graph includes most probably a syntactically correct sentence hypothesis. Thus, the parsing process stops before exhausting all possibilities as soon as it finds the desired parse tree. The N-best list approach performs similar to the first-best approach if the first hypothesis is the correct one. As might be expected, the hypothesis that is picked after a successful parse in any of the approaches does not mean to be the desired hypothesis.

Next, we have conducted an experiment to show the relation between the word-graph size and the number of edges built during processing for each approach. For this purpose, we have picked from newspapers 20 real-life phrases for each word-graph size segment and run through the SR and word-graph pruner with N=10. On the average, the correct hypothesis was at the sixth place. As depicted in Chart 1, the ratio remains the same as the number of nodes in the word-graph increases. In the word-graph processing approach, the total number of edges remains below acceptable boundaries even for big sized word-graphs. For separate processing, however, this number is above practical limits especially in large word-graphs.

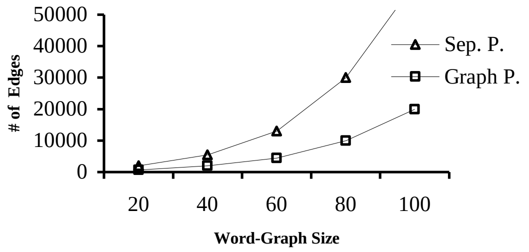

Chart 1. Relation between word-graph size and number of edges (N=10).

4. CONCLUSIONS

In this paper, we present a new approach to couple SR and a rule-based MT for speech translation purpose. This approach can be generalized to any MT system employing chart parsing in its analysis stage. Thus, the extensions proposed in this study can pertain also to statistical MT framework utilizing a chart. However, our focus is on a rule-based MT. Besides utilizing rule-based MT in ST, this study uses word-graphs and chart parsing with new extensions. The experiments described in this paper show that parsing the word-graph at one instance improves the translation performance, compared to parsing all sentence hypotheses separately. For further improvement of the ST system, our future studies include the following:

1. Evaluation of the total ST system based on standard assessment measures (word error rate, BLEU, etc.) It will be interesting to see how our coupling performs compared to other approaches.
2. Supplementing the word-graph pruning module with the language grammar rules to improve the quality of the N-best word-graph.
3. Utilizing the statistical information, coming from the SR module, inside MT towards a hybrid system. This will supplement the ambiguity resolving process in syntactic analysis.

5. ACKNOWLEDGMENTS

This work is partially financed by Applications Technology, Inc., USA. Thanks for Jude Miller and Nagendra Goel for their valuable comments and kind support.

6. REFERENCES

[1] L. Chien, K. Chen, and L. Lee, “A Best-First Language Processing Model Integrating the Unification Grammar and Markov Language Model for Speech Recognition Applications,” *IEEE Transactions on Speech and Audio Processing*, Vol. 1, No.2, pp. 221-240, 1993.

[2] H. Ney, “Speech translation: Coupling of Recognition and Translation,” *Proc. ICASSP*, 1999.

[3] R. Zhang, G. Kikui, “Integration of Speech Recognition and Machine Translation: Speech Recognition Word Lattice Translation,” *Speech Communication*, Vol.48, Issues 3-4, pp. 321-334, 2006.

[4] E. Matusov, S. Kanthak, and H. Ney, “On the Integration of Speech Recognition and Statistical Machine Translation,” *Proc. InterSpeech*, 2005.

[5] S. Saleem, S. C. Jou, S. Vogel, and T. Schultz, “Using Word Lattice Information for a Tighter Coupling in Speech Translation Systems,” *Proc. ICSLP*, 2004.

[6] V. Arranz, et. al., “A Speech-to-Speech Translation System for Catalan, Spanish and English,” *AMTA-2004*, Washington DC.

[7] N. Bertoldi, and M. Federico, “A New Decoder for Spoken Language Translation Based on Confusion Networks,” *IEEE ASRU Workshop*, 2005.

[8] E. Ringger, “A Robust Loose Coupling for Speech Recognition and Natural Language Understanding,” *Technical Report*, The University of Rochester, Computer Science Department, Rochester, New York, 1995.

[9] W. Shen, R. Zens, N. Bertoldi, and M. Frederico, “The JHU Workshop 2006 IWSLT System”, *Proc. IWSLT*, 2006.

[10] M. P. Harper et. al., “Integrating Language Models with Speech Recognition,” *Proc. AAAI Workshop on the Integration of Natural Language and Speech Processing*, pp. 139-146, 1994.

[11] H. Weber, “Time Synchronous Chart Parsing of Speech Integrating Unification Grammars with Statistics,” *Proc. of the 8th Twente Workshop on Language Technology*, Dec. 1994.

[12] C. Boitet, and M. Scligman, “The "Whiteboard" Architecture: A Way to Integrate Heterogeneous Components of NLP Systems,” *Proc. COLING*, 1994.

[13] K. Hall, “Best-First Word Lattice Parsing: Techniques for Integrated Syntax Language Modeling,” *PhD Thesis*, Dept. of Computer Science; Brown University, 2005.

[14] K. Hall, and M. Johnson, “Language Modeling Using Efficient Best-First Bottom-Up Parsing,” *Proc. ASRU*, 2003.

[15] K. Hall, and M. Johnson, “Attention Shifting for Parsing Speech,” *Proc. ACL*, 2004.

[16] C. Yen-Lu and S. Richard, “The N-Best Algorithm: An Efficient Procedure For Finding Top N Sentence Hypotheses,” *Proc. HLT*, 1989.