ROBUST ANALYSIS OF SPOKEN INPUT COMBINING STATISTICAL AND KNOWLEDGE-BASED INFORMATION SOURCES

Roldano Cattoni, Marcello Federico

ITC-irst
I-38050 Povo, Trento, Italy
{cattoni,federico}@itc.it

Alon Lavie

Carnegie Mellon University
Pittsburgh, PA 15213, USA
alavie@cs.cmu.edu

ABSTRACT

The work presented in this paper concerns the analysis of automatic transcription of spoken input into an interlingua formalism for a speech-to-speech machine translation system. This process is based on two sub-tasks, (1) the recognition of the Domain Action (a speech act and a sequence of concepts) and (2) the extraction of arguments consisting of feature-value information. Statistical models are used for the former, while a knowledge-based approach is employed for the latter. This paper proposes an algorithms that improves the analysis in terms of robustness and performance: it combines the scores of the statistical models with the extracted arguments, taking in account the well-formedness constraints defined by the interlingua formalism.

1. INTRODUCTION

This paper presents recent work carried out at ITC-irst in the framework of a speech-to-speech machine translation project, named NESPOLE![1]. Automatic speech translation is applied to a tourism domain, and builds on the interlingua approach already experimented within the C-STAR consortium[2]. This work, in particular, focuses on the so called analysis step, that is designed to map the automatic transcription of an utterance into its interlingua representation. This process requires: (1) segmenting the utterance into semantic dialogue units (SDUs); (2) recognizing the Domain Action (DA), consisting of a speech act and a sequence of concepts, expressed in each individual SDU; and (3) extracting arguments, consisting of feature-value information. In addition to the extraction of arguments, performed by means of hand designed recursive transition networks, the other steps are performed by applying statistical models. In particular, the speech act and concept extraction exploits statistical classifiers based on n-gram language models. This work shows that improvements on the analysis step can be achieved by integrating scores of the statistical models with well-formedness constraints defined by the interlingua formalism. More precisely, we present an algorithm that tries to find the optimal interlingua representation by taking into account the scores provided by the statistical classifiers, the arguments matched by the parser, and the rules defining well-formed interlingua representations.

This paper is arranged as follows. Section 2 introduces the NESPOLE! project and the used interlingua formalism, Section 3 describes the implemented analysis module. Finally section 4 presents and discusses experimental results obtained on a set of 11 Italian dialogues.

2. INTERLINGUA-BASED MACHINE TRANSLATION IN THE NESPOLE! PROJECT

NESPOLE! is a speech-to-speech machine translation research project funded jointly by the European Commission and the US NSF. The main goal of the NESPOLE! project is to advance the state-of-the-art of speech-to-speech translation in a real-world setting of common users involved in e-commerce applications. The project is a collaboration between three European research labs (ITC-irst in Trento Italy, ISL at University of Karlsruhe in Germany, CLIPS at UJF in Grenoble France), a US research group (ISL at Carnegie Mellon in Pittsburgh) and two industrial partners (APT - the Trentino provincial tourism bureau, and AETHRA - an Italian tele-communications commercial company). The speech-to-speech translation approach taken by the project builds on previous work that the research partners conducted within the context of the C-STAR consortium. The prototype system developed in NESPOLE! is intended to provide effective multi-lingual speech-to-speech communication between all pairs of four languages (Italian, German, French and English) within broad, but yet restricted domains. The first showcase currently under development is in the domain of tourism and travel information.

NESPOLE! uses an interlingua-based approach with a relatively shallow task-oriented interlingua representation [1],

[1] NESPOLE! - NEgotiating through SPOken Language in E-commerce. See the project web-site at http://nespole.itc.it/

[2] See the consortium web-site at http://www.c-star.org

that was initially designed for the C-STAR consortium and has been significantly extended for the NESPOLE! project. Interlingual machine translation is convenient when more than two languages are involved because it does not require each language to be connected by a set of transfer rules to each other language in each direction [2]. Adding a new language that has all-ways translation with existing languages requires only writing one analyzer that maps utterances into the interlingua and one generator that maps interlingua representations into sentences. The interlingua approach also allows each partner group to implement an analyzer and generator for its home language only.

The NESPOLE! interlingua, called Interchange Format (IF), consists of four representational components: (1) the speaker tag ("*a:*" stands for agent, "*c:*" for client); (2) the speech act (e.g. *thank*, *give-information*); (3) a possibly empty sequence of concepts, describing the focus (e.g. *+hotel*, *+room*); (4) a possibly empty list of arguments as name-value pairs (e.g. *room-type=double*). The following are three examples of utterances tagged with their corresponding IF label:

1. `Thank you very much`
   c:thank
2. `And we'll see you on February twelfth`
   a:closing (time=(february, md12))
3. `There is an hotel in the town`
   a:give-information+existence+accommodation (accommodation-spec=hotel, location=town)

3. INTERCHANGE FORMAT COSTRUCTION

This section focuses on the analysis module developed for the Italian component of the NESPOLE! system. The first step of analysis involves semantic segmentation: the automatic transcription provided by the acoustic recognizer is split into semantic segments called *Semantic Dialogue Units*, or SDUs). In the second step of analysis, a single interlingua representation (IF) is assigned to each SDU. The output of the analysis module is therefore a (possibly unary) sequence of IF representations. The speaker tag of each IF is easily determined: it is assigned a-priori depending on the role played by the person, agent or client. The other IF components (speech-act, concepts and arguments) are determined by analysis of the SDU.

In the current implementation, speech act and concepts are treated together: the string obtained by concatenating the speech act and the (possibly empty) sequence of concepts is considered as a single label corresponding to the Domain Action (DA). Given a SDU, determining speech act and concepts is therefore approached as a classification problem, that is finding the "best" label among those encoding the allowed Domain Actions. A statistical technique based on language models is employed to classify DAs; it is described in subsection 3.1.

For argument extraction, a knowledge-based approach is used: a recursive transition network (RTN) parser based on semantic grammars outputs a sequence of parse trees, semantically corresponding to different IF arguments. Such trees are then mapped into the appropriate IF syntax. Subsection 3.2 provides details on the developed technique.

Decomposing the problem of IF construction into two separated and independent sub-tasks, Domain Action classification and argument extraction, has several advantages (reduced complexity, specialized techniques for the sub-tasks). However the main drawback is that the information extracted by the separated sub-tasks may be not consistent when combined together, resulting in illegal IFs. Moreover a sub-task cannot exploit the information extracted by the other in order to improved its performance.

The issue of the legality of the produced IFs is a crucial one. By definition each item in a Domain Action (the speech act and concepts) licenses a set of arguments that are semantically related to the item. For example the concept *+accommodation* licenses several arguments that encode the type, class, board of the accommodation, its location and so on. An IF in which there are arguments not licensed by at least one of the items in the Domain Action is illegal – its meaning cannot be determined and a meaningful target language sentence cannot be generated from such an IF. Let us introduce an instance in which an illegal IF may be produced. Given the utterance of the third example in section 2, its correct IF label is

> *a:give-information+existence+accommodation (accommodation-spec=hotel, location=town)*

where *give-information+existence+accommodation* encodes the DA, and *(accommodation-spec=hotel, location=town)* encodes the arguments. However, due to the statistical approach used for Domain Action classification, the predicted Domain Action (that has maximum likelihood) may in fact be incorrect, due to either speech recognition errors or a simple misclassification. In the above example, the classifier may select the DA label of *introduce-self*, that licenses only arguments related to a person introducing himself. In this case the pure composition of the selected DA label and the extracted arguments produces the following IF

> *a:introduce-self (accommodation-spec=hotel, location=town)*

which is illegal: neither *accommodation-spec=* nor *location=* are licensed by *introduce-self*, and there is no way to attribute a meaning to this IF. In order to overcome such problems we have developed the algorithm described in subsection 3.3.

3.1. Domain Action classification

The IF specifications define 57 speech-acts and 98 concepts; some of the DAs are specific to the travel domain, while others represent DAs that are not domain-specific (such as greetings, communication acts, etc.). The number of legal Domain Actions obtained by concatenating appropriate speech-acts and concepts is rather high (several thousands). However, the number of DAs that are commonly used in actual dialogues is substantially smaller (fewer than 300).

For the classification of Domain Actions an approach based on language models is used. For a given SDU, the selected DA is the one corresponding to the language model that provides the highest likelihood. To reduce the problem of data sparseness, a labeling pre-processing is performed on the SDUs. Some text words in the utterance are substituted with labels corresponding to classes of semantically equivalent words or expressions such as greetings, hotel names, locations, etc. Bigram language models were estimated for each DA by using the smoothing method described in [3].

3.2. Argument extraction

The IF specifications define 140 top-level arguments, some of which may include sub-arguments. For argument extraction, a knowledge-based approach in two steps is used: first a RTN parser is applied on the pure text; it produces a sequence of parse trees, semantically corresponding to different IF arguments. In the second stage, the parse trees are converted into their appropriate IF syntax.

Parsing is performed by applying the ITC-irst HMM decoder [4] on the input text (rather than on an acoustic signal). Arguments are thus modeled with recursive finite state networks, which represent, according to the case, word lists (e.g locations, digits), regular expressions (e.g simple temporal expression, integers), or bigram language models (e.g complex temporal expressions). In particular, complex expressions can be expressed in terms of more simple ones using recursion. For the current NESPOLE! domain we have developed about 407 grammars. Although most of arguments are domain-dependent (e.g room and hotel) there are also many cross-domain arguments (e.g numbers, price or temporal expressions). The output of the HMM decoder is a sequence of parse trees corresponding to the most probable path trough the recursive finite state networks defined by the grammars and language models. A rule-based procedure written in Perl is then used to map the parse trees into IF-compliant arguments.

3.3. Combining Domain Action classification and argument extraction

As described earlier, constructing the IF by simply composing the outputs of the Domain Action classifier and argument extractor is prone to the production of illegal IFs. Moreover there is no exchange of information between the two processes that may improve the quality of their performance. The algorithm we have developed to overcome such limitations exploits the well-formedness information contained in the IF specifications, in particular the relationship between speech-acts/concepts and their licensed arguments.

The first step of the algorithm is the argument extraction, in the same way as described in subsection 3.2. In addition a list with the top-level argument names is produced. For example from the sentence "`I would like to take a trip to Italy`" the extracted arguments are *(disposition=desire, visit-spec=trip, location=italy)*, so the list of top-level argument names is {*disposition= visit-spec= location=*}.

In the second step each argument name is associated with all the Domain Action items (speech-acts and concepts) that can license it, according to the IF specifications. In our example this associative table is

Argument Name	DA Items Licensing It
disposition=	*+disposition*
visit-spec=	*+package +trip*
location=	*greeting +accommodation .. +view*

The *disposition=* argument can be licensed only by the concept *+disposition*: this means that this argument is very discriminant. For *visit-spec=* there are two licensing concepts, *+package* and *+trip*, corresponding to two different contexts in which the argument may appear. There are several items (both speech-acts and concepts) that license *location=*: in fact, locative expressions are cross-domain and may appear in almost all contexts.

In the third step an extended version of the Domain Action classifier is performed taking into account the associative table in addition to the labelled text. For each DA, two numbers are computed. First, its probability given by its language model. Second, the number of matched arguments for the particular DA label. Each argument is considered matched when the argument is licensed by at least one of the items contained in the DA, according to the associative table constructed for the arguments. In our above example, the number of matches of the DA *give-information+disposition+accommodation* is 2, since *disposition=* and *location=* are licensed respectively by *+disposition* and *+accommodation*; *visit-spec=* is not licensed by any of *give-information*, *+disposition* or *+accommodation*. In the case of the DA *give-information+disposition+trip* all three arguments are matched since each of the arguments has at least one licensing item in the Domain Action.

In the final stage of the extended algorithm, the selected DA is the one having the highest language model probability among those with the highest number of matched arguments. Before constructing the final IF label, a filtering step is performed. If the number of matched arguments is less than the total number of arguments, the arguments that are not licensed by the DA are identified and removed from the

IF – note that since we select a DA that has the maximal number of matched arguments, this implies that no DA can in fact license all of the extracted arguments. The filtering thus guarantees that the produced IF is legal.

4. EVALUATION

We compared the performance of the extended algorithm (labelled *combined* in Table 1) with the previous analysis algorithm (labelled *separated*) via a cross-validation evaluation test. Since the corpus contains 11 annotated dialogues, at each step one dialogue was selected for the test set and the other ten dialogues for the training set. At the end of the 11 steps, the average statistics were calculated.

Table 1 reports a comparison of the performance of the two algorithms with respect to the speaker side (agent, client or both). The second column contains the number of the SDU in the test. The third and fourth columns reports statistics on two partial views of the produced IFs, respectively the percentage of correctly classified Domain Actions (independently of the extracted arguments) and the percentage of legal IFs (independently of their correctness). The last column is the most meaningful in terms of overall performance: it reports the percentage of correct (that is legal and with a correctly classified Domain Action) IFs.

SEPARATED				
Speaker	*# SDU*	*correct DA*	*legal IF*	*correct IF*
a(gent)	775	53.2 %	72.9 %	45.9 %
c(lient)	512	58.8 %	85.0 %	57.6 %
a+c	1287	55.4 %	77.7 %	50.6 %

COMBINED				
Speaker	*# SDU*	*correct DA*	*legal IF*	*correct IF*
a(gent)	775	50.5 %	100.0 %	50.5 %
c(lient)	512	61.1 %	100.0 %	61.1 %
a+c	1287	54.7 %	100.0 %	54.7 %

Table 1. Results of the cross-validation test with *separated* and *combined* algorithm. The last column on the right shows the global performance in terms of percentage of correct IFs produced: a significant improvement was obtained with the combined algorithm.

In term of overall performance it can be seen that the combined algorithm performs substantially better than the separated one (a relative performance gain of more than 8% was observed). It is interesting to note that the pure Domain Action classification performs slightly better in the separated algorithm: but the high percentage (almost 1/4) of illegal IFs it produces significantly reduces the overall performance. The cases in which the combined algorithm does not correctly classify the DA (while the separated algorithm is correct) correspond to cases in which the extracted arguments are not completely correct. In these cases the combined algorithm is "driven away" from the correct DA towards other DAs that better match the extracted arguments. However, we have observed that, even in these cases, the combined algorithm often produces reasonable (although not completely correct) IFs. This is a result of both the removal of poorly identified arguments as well as well as the occasional selection of a semantically more general DA.

This effect of the combined algorithm relates to the issue of the robustness of our analysis approach. Both DA classification and argument extraction are prone to make errors due to several different factors. Some of these are internal to the analysis module (e.g. too few training examples for the language models, imperfect grammars, poor segmentation in SDUs) while other factors are external (e.g. wrong words reported by the acoustic recognizer). As explained above, in such cases the combined algorithm tends to produce IFs that are reasonable – in the worst case partial – but never illegal. This behavior has a positive impact on the practical use of the dialogue translation system: users typically accept imperfect/partial but reasonable translations since the dialogue can go on with little interruption. On the other hand, users often are confused and frustrated when there is no translation for their utterance due to illegal IFs.

Another advantage of the proposed algorithm is that it supports self-debugging during system development. By tracing the matches between arguments and Domain Actions it is easier to highlight errors in the argument extraction process (e.g. typos in grammars), or the lack of relevant examples in the Domain Action classifier.

5. REFERENCES

[1] L. Levin, D. Gates, A. Lavie, and A. Waibel, "An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues," in *Proc. of the International Conference on Spoken Language Processing*, Sydney, Australia, 1998, pp. Vol. 4, 1155–1158.

[2] S. Nirenburg, J. Carbonell, M. Tomita, and K. Goodman, *Machine Translation: A Knowledge-Based Approach*, Morgan Kaufmann, San Mateo, CA, 1992.

[3] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inform. Theory*, vol. IT-37, no. 4, pp. 1085–1094, 1991.

[4] F. Brugnara and M. Federico, "Dynamic language models for interactive speech applications," in *Proc. of the 5th European Conference on Speech Communication and Technology*, Rhodes, Greece, 1997, pp. 2751–2754.