A Pattern-Based Analyzer for French in the Context of Spoken Language Translation: First Prototype and Evaluation

Hervé BLANCHON
GETA-CLIPS (IMAG)
BP 53
38041 Grenoble Cedex 9, France
herve.blanchon@imag.fr

Abstract

In this paper, we describe a first prototype of a pattern-based analyzer developed in the context of a speech-to-speech translation project using a pivot-based approach (the pivot is called IF). The chosen situation involves a French client talking to an Italian travel agent (both in their own language) to organize a stay in the Trentino area.

An IF consists of a dialogue act, and a list, possibly empty, of argument values. The analyzer applies a "phrase spotting" mechanism on the output of the speech recognition module. It finds well-formed phrases corresponding to argument values. A dialogue act is then built according to the instantiated arguments and some other features of the input.

The current version of the prototype has been involved in an evaluation campaign on an unseen corpus of four dialogues consisting of 235 speech turns. The results are given and commented in the last part of the paper. We think they pave the way for future enhancements to both the coverage and the development methodology.

Résumé

Dans cet article, nous décrivons la première version d'un analyseur fondé sur des patrons dans le contexte d'un projet de traduction de parole utilisant une technique de traduction par pivot (le pivot est appelé IF). Dans la situation choisie, un client français parle avec un agent italien (chacun dans sa langue maternelle) pour organiser un séjour dans la région du Trentin en Italie.

Une IF se compose d'un acte de dialogue et d'une liste, éventuellement vide, de valeurs d'arguments. L'analyseur met en œuvre un mécanisme de reconnaissance de syntagmes sur la sortie du module de reconnaissance de la parole. Cela permet de trouver des syntagmes bien formés qui correspondent à des valeurs d'arguments. L'acte de parole est alors construit en utilisant les arguments instanciés ainsi que d'autres caractéristiques de l'entrée.

Cette version du prototype a été mis en œuvre lors d'une évaluation sur un corpus de quatre dialogues, non utilisés pour le développement, composé de 235 tours de parole du client. Les résultats sont donnés dans la dernière section de cet article. Nous pensons qu'ils ouvrent la voix pour de futures améliorations de la couverture ainsi que de la méthodologie de développement.

Introduction

In the framework of the NESPOLE! project [Besacier, L., *& al.*, 2001; Lazzari, G., 2000] funded by the EU and the NSF we are exploring future applications of automatic speech-to-speech translation in the e-commerce and e-services areas. For the actual translation we are using a pivot-based approach (the pivot is called IF for Interchange Format). Thus, we have to develop the analysis from the textual output of an automatic speech recognition module towards the IF and the generation from the IF towards a text-to-speech input text.

In this context, the analyzer has to be robust against ill-formed input (in terms of syntax) and recognition errors, both likely to be quite common. To cope with these problems several families of methods may be used: a rule-based approaches with rules being relaxed if needed, a “let the number do every thing” approaches (using aligned source language inputs and their pivot representations), a pattern-based approaches (focusing on important features of the input), and finally a mixture of the previous ones.

Taking into account the way the pivot represents the information present in the input and the possible methods, we chose to investigate a pattern-based approach (as in [Zong, C., *& al.*, 2000]). In this paper we will focus on the first prototype of an analysis module from French to a pivot called IF (Interchange Format). We will justify our choices with regard to the pivot specification and describe the realization; finally we will give several numbers about an evaluation this analyzer was involved in.

1 Context and choices

1.1 Interchange Format

The IF we are currently using is an extension of the one used in the C-STAR II context [Levin, L., *& al.*, 2000; Levin, L., *& al.*, 1998]. It is designed to abstract away from peculiarities of any particular language in order to allow for translation that are non-literal but capture the speaker’s intent.

The IF is based on domain actions (DAs) that consist of speech act and concepts. We have currently defined 62 general speech acts (e.g. acknowledge, introduce-self, give-information,). The concepts are split into 9 attitudes (e.g. disposition, feasibility, obligation), and 97 main predications or predication participants (e.g. price, room, activity).

In addition to the DA, an IF representation may contain arguments (e.g. disposition, price, room-spec). The arguments have values that represent information about the speech acts and the concepts. There is currently about 280 arguments, 150 of them being top-level (e.g. disposition, price, room-spec). The others arguments do not exist on their own, they are embedded within the top-level arguments (e.g. quantity, currency, identifiability).

For an utterance meaning "je voudrais la chambre à 70 euros"[1] the IF would be:

```
c:give-information+disposition+price+room
  ( disposition=(who=i, desire),
    price=(quantity=70, currency=euro),
    room-spec=(identifiability=yes, room)
  )
```

c: indicates that the client is speaking. give-information+disposition+price+room is the DA. disposition, price, room-spec are the top-level arguments. quantity, currency, identifiability are embedded arguments.

1.2 Constaints

The IF specification is giving constraints on the construction of an IF at each levels.

Speech acts are defined with their possible concept continuations (e.g. disposition, price, availability concepts may follow give-information) and their licensed arguments (rhetorical relations, e.g. cause, conjunction, disjunction). Concepts are also defined with their possible continuations (e.g. accommodation, room, activity concepts may follow price) and arguments (e.g. for-whom, price, time arguments may be arguments of price).

Arguments are defined by their possible value, relations and attributes. The value may be question (the argument is questioned) or a set of actual values (e.g. double, single, twin for a room-spec). It is also possible to handle relatives and pronouns. Relations define links between two concepts (e.g. bed-spec, location, price for a room-spec). Attributes define links between a concept and a set of values (e.g. quantity, identifiability). An attribute is defined only with a value and attributes, no relation.

1.3 Choices

We had to choose a methodology for the analysis from a speech recognition output towards an IF and the generation form an IF to a French text.

For the generation it was decided to apply a rule-based approach under Ariane-G5 [Boitet, C., 1997]. We are using the IF specification files to

[1] "I would like the room that costs 70 euros"

automatically produce parts of the dictionaries and the grammars. The IF is parsed into a French linguistic tree passed to a general-purpose French generation module. This approach forces us to develop a specification as clean as possible describing all the possible "events". At the end, every potential IF input should be covered. The drawbacks of this approach are the bootstrapping process, which takes a huge amount of time, and the continuous changes in the IF specification we have to cope with. For the generation, we are also experimenting a pattern-based approach in which DA families are associated with template sentences (a fill in the blanks sentence). If possible, the blanks are filled, with the French phrase associated with the right argument value.

For the analysis we are also pursuing two tracks. When the generation module will be available we are thinking of reversing the process to build an analysis module (an analyzer realized for C-STAR II is described in [Blanchon, H. and Boitet, C., 2000, Boitet, C. and Guilbaud, J.-P., 2000]). We are also using a pattern-based approach that is very convenient to deal with the output that may be produced by the speech recognition module. The operation of the analyzer can be viewed as "phrase spotting" among the input so that insertion, deletion, and wrong agreements and word can be dealt with. The source code is written in Tcl with an intensive use of regular expressions matching. The next section is dedicated to this module.

2 Overall process

The French to IF process is divided in four main steps. The input text is first split into semantic dialogue units (SDUs[2]). The topic of each SDU is then searched out. According to the topic, the possible arguments are then instantiated. Finally, the Dialogue Act is built using the instantiated arguments and some other features of the SDU.

2.1 Turns splitting into SDU

To split the output of the ASR we are using boundaries: simple phrases and articulations. Simple phrases give speech acts without continuation. They are classified into:

- affirmations (e.g. oui c'est ça, bien sûr)[3],
- negations (e.g. non pas du tout, non pas très bien, non)[4],
- acknowledgments (e.g. c'est d'accord, c'est ok, c'est bien, ok, oui)[5],
- apologies (e.g. excusez-moi, désolé, pas de quoi)[6],
- exclamations (e.g. c'est excellent, très bien, oh)[7],
- greetings (e.g. bonjour, au revoir, bonne journée)[8],
- dialogue management (e.g. allô, j'entends, j'écoute)[9],
- thanks (e.g. merci bien, merci)[10], and some others.

Articulations are realizations of the rhetorical arguments (e.g. simple conjunctions, conjunction phrases) followed by a pronoun or the beginning of a question (e.g. et donc je, et puis il, et j', donc on, est-ce que, quel est)[11].

Some thirty regular expressions are used for the splitting. Examples are given in annex 1.

2.2 Topic detection

Here, the goal is to find either the terminal speech act for the SDU (e.g. `apologize`, `contradict`, `exclamation`, `greeting`) or what is the SDU talking about (e.g. `accommodation`, `room`, `activity`, `attraction`, `price`).

A list of expressions and/or words is associated with the terminal speech acts (e.g. `bonjour`, `salut`, `à bientôt`, `bonsoir`, `au revoir`, `enchanté`, `à plus tard`, `à plus`, `bonne journée` for the `greeting`)[12] and with the

[2] An SDU is a part of an utterance that can be analyzed into an unique IF.

[3] yes that it, of course

[4] no not at all, no not very well, no

[5] it's ok, it's ok, that's right, ok, yes

[6] excuse me, sorry, you are welcome

[7] that's excellent, very good, oh

[8] hello, good bye, have a good day

[9] hello, I can hear, I am listening

[10] thank you very much, thank you

[11] and thus I, and then I, and I, so we, is …, what is

[12] hello, hi, see you soon, good evening, good bye, delighted, see you later, see you later (casual), have a good day

other topics (e.g. place de camping, salle de conférence, chambre double, chambre simple, suite, … for room)[13].

The instantiated topic is the first matched one in the utterance. If there is no match, the topic is set to “unknown”. The latter concerns fragments with no explicit topic given (e.g. an utterance containing only "1 3 5 7") or topics not handled yet. There are currently 30 topics defined.

2.3 Arguments filling

A Topic2If function is then in charge of finding the instantiated arguments among the possible ones for the given speech act and/or topic. For example, for the room topic some of the possible arguments are room-spec, location, and duration.

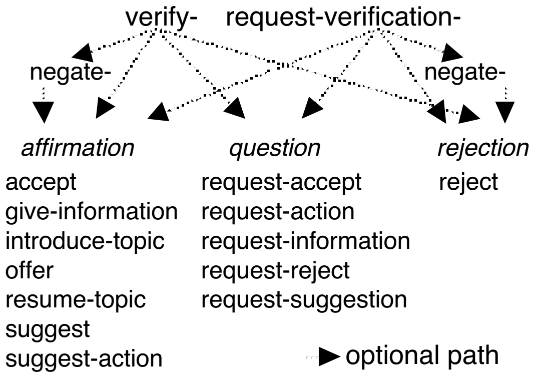
An argument filling function (Argument2If) is associated with each defined argument. Those functions are in charge of finding a possible realization for its argument in the input. It takes into account the value, the relations and attributes. For example for a room-spec the RoomSpec2If function (given in annex 2) is trying to locate an identifiability, a quantity and a value (the type of room). This is done by trying to match a sequence made of an identifier (e.g. une, un, des, plusieurs)[14], a number and a room type. The smallest acceptable sequence being a room-type. If found, the French room type is translated into an IF room type through the RoomSpec2If function defined in the fifservdico space name. The result is the IF-encoded value of the argument or an empty string.

Up to now we are covering one fifth of the top-level arguments (32 over 150, the most common ones) given that the 21 rhetorical arguments and 8 out of the 9 attitude arguments are not handled yet. Considering the total number of arguments one fourth of them (73 over 281) is handled, knowing that among the 281 arguments, there is 35 synonymous definitions.

2.4 Dialogue act construction

The dialogue act is built concatenating the speech act, the attitudes (currently disposition only), the main predication and the predication participants.

The speech act is calculated using information about – the verbal construction (affirmation, question, and rejection), – the potential negation of the predicate, – the potential verification or request for verification of the predicate.



If present, the attitude is recognized with cue phrases. The Disposition2If function is given in annex 3.

Then, according to the matched arguments, the main predication and predication participant are calculated.

Finally the IF is built by concatenating the speaker (a: or c:), the dialogue act and the arguments values.

3 Evaluation

The analyzer has not been evaluated on its own yet (grading the IFs produced). We have performed a set of end-to-end evaluations on the translation chain (analysis and generation) that give some information on the performances of the analyzer itself. The generator, being developed in parallel with the analyzer, covers the IFs produced by the analyzer.

3.1 Evaluation suites

We performed both mono-lingual evaluation, as well as cross-lingual evaluation We evaluated on both textual manually transcribed input as well as on input from actual speech-recognition of the original audio.

We graded the word accuracy rate obtained by the recognition engine. We also graded the speech recognized output as a "paraphrase" of

[13] camping lot, conference room, twin room, single room, suite

[14] a (feminine), a (masculine), some, several

the transcriptions, to measure the semantic loss of information due to recognition errors.

In the following we will give the results for mono-lingual French-to-French evaluation. The whole set of results is given in [Lavie, A. *& al.* 2002]

3.2 Data and Grading

3.2.1 Data set

The French data set was made of four dialogs extracted from the NESPOLE database [Burger & al. 2000]. Two of them were related to a client / agent discussion for organizing winter holidays in Val di Fiemme in Italy; the two others were related to summer vacations in the same region. Speech signals were then re-recorded from client turn transcriptions of these 4 dialogs (8kHz sampling rate). This data represents 235 signals related to 235 speaker turns of two different speakers (1 male, 1 female). Finally, these 235 speaker turns were segmented manually into 427 SDUs for translation evaluation. These turns were also segmented automatically into 407 SDUs by the SDU segmentation step of the analyzer. We had thus 2 sets of SDUs.

3.2.2 Grading

All graders then used these segmentations in order to assign scores for each SDU present in the utterance. We followed the three-point grading scheme previously developed for the C-STAR consortium, as described in [Levin, L. *& al*. 2000]. Each SDU is graded as either *Perfect* (meaning translated correctly and output is fluent), *OK* (meaning is translated reasonably correct but output may be disfluent), or *Bad* (meaning not properly translated). We calculate the percent of SDUs that are graded with each of the above categories. *Perfect* and *OK* percentages are also summed together into a category of *Acceptable* translations. Average percentages are calculated for each dialogue, each grader, and separately for client and agent utterances. We then calculated combined averages for all graders and for all dialogues for each language pair.

3.3 Results

In the following tables the result are given for acceptable paraphrase (for the ASR) and acceptable translation (for monolingual and cross lingual translation).

3.3.1 Results on automatic SDUs

Monolingual Translation

Language	Transcribed	Speech Rec.
French-to-French	62%	48%

Table 1: French Monolingual End-to-End Translation Results (Percent Acceptable) on Transcribed and Speech Recognized Input on Analyzer's SDUs

3.3.2 Results on manual SDUs

Speech Recognition

Language	WARs	Acceptable Paraphrase
French	71.2%	65.0%

Table 2: Speech Recognition Word Accuracy Rates and Results of Human Grading (Percent Acceptable) of recognition Output as a Paraphrase

Monolingual Translation

Language	Transcribed	Speech Rec.
French-to-French	54%	41%

Table 3: Monolingual End-to-End Translation Results (Percent Acceptable) on Transcribed and Speech Recognized Input

3.4 Comments

On Speech Recognition

About 65% of the SDUs were judged correctly paraphrased. This score is more informative than the WAR% since it means that 35% of the SDUs will not be correctly translated. This evaluation is also a good way to check that the graders give more or less the same scores (it is the case here).

On French-to-French Monolingual Translation

About 54% of the SDUs were judged acceptably translated on the transcribed data. Thus, we know for sure that 46% of the SDUs will not be correctly translated anyway. It is thus important to know if this percentage includes the SDUs badly recognized by the ASR system or not. That is shown in the next paragraph.

About 41% of the SDUs were judged acceptably translated on the speech recognized data. This result alone would have been very difficult to interpret, but the previous results show the respective contribution of ASR and Translation to this performance. It is an important information that will be used to further improve the system.

On the Results on Automatic SDUs

The results we are producing here are isolated as the same experiment has not been done for the other languages. The results for French-to-French are better, +7%. The number of automatically produced SDUs (407) is lower than the number of manually produced SDUs (427). A careful study of the results shows that the *perfect* scores are quite the same, but the number of *OK* scores is higher with the automatically produced SDUs. However, the data have to be checked carefully to give a grounded conclusion.

This phenomenon, if it is confirmed by our partners, may explain part of the fact that user studies and system demonstrations indicate that while the current level of translation accuracy cannot be considered impressive, it is already sufficient for achieving effective communication with real users.

Conclusion

In this paper we have described our first evaluated prototype of a pattern-based analyzer from spoken French into IF.

We have tried to show that this approach seems very promising. The "phrase spotting" mechanism we implemented handle quite well the output of a speech recognizer whose language models allows for the construction of isolated correct segments.

Having a unique construction method for each argument and a common DA construction pattern shared between the topics allows for reusability and easy updating of the code to follow the regular IF specification changes.

The first results we have shown are encouraging and pave the way for better results in the next NESPOLE! evaluation studies.

Acknowledgements

Our thanks go to the graders (L. Besacier, D. Vaufreydaz, S. Mazenot, S. Rossato, A.C. Descalle), and to the CLIPS and IRST teams for their contribution to this paper.

References

Besacier, L., Blanchon, H., Fouquet, Y., Guilbaud, J.-P., Helme, S., Mazenot, S., Moraru, D. and Vaufreydaz, D. (2001) *Speech Translation for French in the NESPOLE! European Project.* Proc. Eurospeech. Aalborg, Denmark. September 3-7, 2001. vol. 2/4: pp. 1291-1294.

Blanchon, H. and Boitet, C. (2000) *Speech Translation for French within the C-STAR II Consortium and Future Perspectives.* Proc. ICSLP 2000. Beijing, China. Oct. 16-20, 2000. vol. 4/4: pp. 412-417.

Boitet, C. (1997) *GETA's metodology and its current development towards networking communication and speech translation in the context of the UNL and C-STAR projets.* Proc. PACLING-97. Ome, Japan. 2-5 September, 1997. vol. 1/1: pp. 23-57.

Boitet, C. and Guilbaud, J.-P. (2000) *Analysis into a Formal Task-Oriented Pivot without Clear Abstract Semantics is Best Handled as "Usual" Translation.* Proc. ICSLP 2000. Beijing, China. Oct. 16-20, 2000. vol. 4/4: pp. 436-439.

Burger, S., Besacier, L., Metze, F., Morel, C. and Coletti, P. (2001) *The NESPOLE! VoIP Dialog Database.* Proc. Eurospeech. Aalborg, Denmark. September 3-7, 2001.

Lazzari, G. (2000) *Spoken Translation: Challenges and Opportunities.* Proc. ICSLP 2000. Beijing, China. Oct. 16-20, 2000. vol. 4/4: pp. 430-435.

Levin, L., Gates, D., Lavie, A., Pianesi, F., Wallace, D., Watanabe, T. and Woszczyna, M. (2000) *Evaluation of a Practical Interlingua for task-Oriented Dialogue.* Proc. Workshop of the SIG-IL, NAACL 2000. Seattle, Washington. April 30, 2000. 6 p.

Levin, L., Gates, D., Lavie, A. and Waibel, A. (1998) *An Interlingua Based on Domaine Actions for Machine Translation of Task-Oriented Dialogues.* Proc. ICSLP'98. Syndney, Australia. 30th November - 4th December 1998. vol. 4/7: pp. 1155-1158.

Lavie A., Metze F., Pianesi F., Burger S., Gates D., Levin L., Langley C., Peterson K., Schultz T., Waibel A., Wallace D., McDonough J., Soltau H., Laskowski K., Cattoni R., Lazzari G., Mana N., Pianta E., Costantini E., Besacier L., Blanchon H., Vaufreydaz D., Taddei L. (2002) *Enhancing the Usability and Performance of Nespole! – a Real-World Speech-to-Speech Translation System.* Proc. HLT 2002. San Diego, California (USA). March 22-27, 2002. 6p.

Zong, C., Huang, T. and Xu, B., (2000). *An Improved Template-Based Approach to Spoken Language Translation.* Proc. ICSLP 2000. Beijing, China. Oct. 16-20, 2000. vol. 4/4: pp. 440-443.

Annexes

Annex 1: Regular expression for splitting into SDUs

In the following extract of the `SplitHypo` procedure, a sequence made of a simple sentence (leading to an if), a non-empty string, another simple sentence, and a possibly empty string is searched.
If such a sequence is found, the first simple sentence is an SDU (*`theSDU1`*), the non-empty string is split into SDUs, the second simple sentence is an SDU (*`theSDU2`*), and the possibly empty string is split into SDUs.

```
proc SplitHypo {inWho inString} {
  ...
  #simple sentence
  } elseif {[regexp "^ ($simplesentences) (.+?) ($simplesentences) (.*)"
                    $inString lMatch lFirst lSecond lThird lFourth]!=0} {
        append theSDU1 "{<" $inWho "> " $lFirst " }"
        append the_rest1 " " $lSecond " "
        append theSDU2 "{<" $inWho "> " $lThird " }"
        append the_rest2 " " $lFourth
        concat $theSDU1 [SplitHypo $inWho $the_rest1] $theSDU2 [SplitHypo $inWho $the_rest2]
    }
  ...
}
```

Annex 2: `room-spec` argument value construction

In the following extract of the `RoomSpec2If` procedure, a sequence made of a number and a room specification expressed in French eventually preceded by the French plural definite article les is searched.
If such sequence is found then the IF argument
`room-spec=(identifiability=`*yes/no*`, quantity=`*quantity*`,` *room_specification*`)` is constructed whether the article is found or not.

```
proc RoomSpec2If {inString} {
  ...
  } elseif {[regexp "(?:les) (\[0\-9\]+) ($fifservdico::frenchroomspec)(?:x|s)?"
                    $inString lMatch lQuantity lRoomSpec]!=0} {
        append the_result "room-spec=(identifiability=yes, quantity=" $lQuantity ", "
                          [fifservdico::RoomSpec2If $lRoomSpec] ")"
  } elseif {[regexp "(\[0\-9\]+) ($frenchroomspec)(?:x|s)?"
                  $inString lMatch lQuantity lRoomSpec]!=0} {
        append the_result "room-spec=(identifiability=no, quantity=" $lQuantity ", "
                          [fifservdico::RoomSpec2If $lRoomSpec] ")"
  }
  ...
}
```

Annex 3: `disposition` argument value construction

In the following extract of the `Disposition2If` procedure, a sequence made of a French pronoun and a disposition verb eventually surrounded by negation markers is searched.
If such sequence is found then the IF argument **`disposition`**`=(who=`*pronoun*`,` *disposition_verb*`)` is constructed whether the verb is negated of not.

```
proc Disposition2If {inString} {
  if {[regexp "($frenchpronoun) ?(ne |n')?($frenchdispositionverb) (pas)?"
               $inString lMatch lPron lNe lVerb lPas]!=0} {
    if {$lNe!="" || $lPas!="" } {
          append the_result "disposition=(who=" [fifservdico::NormalizePronoun2If $lPron] ", "
                            [fifservdico::NegativDisposition2If $lVerb] ")"
    } else {
          append the_result "disposition=(who=" [fifservdico::NormalizePronoun2If $lPron] ", "
                            [fifservdico::PositivDisposition2If $lVerb] ")"
            }
    }
  ...
  } else {
          return ""
    }
}
```