

Disfluency and Out-Of-Vocabulary Word Processing in Arabic Speech Understanding

Younès Bahou

Lamia Hadrich Belguith

Abdelmajid Ben Hamadou

LARIS - MIRACL Laboratory

Faculty of Economic Sciences and Management of Sfax, B.P. 1088, 3018 - Sfax – TUNISIA

Phone (216) 74 278 777, Fax (216) 74 279 139

bahou_younes@yahoo.fr l.belguith@fsegs.rnu.tn abdelmajid.benhamadou@isimsf.rnu.tn

Abstract

The disfluencies inherent in spontaneous speaking and out-of-vocabulary words omnipresent in any transcribed oral utterance by speech recognition, are a real challenge for speech understanding systems. Thus, we propose in this paper, a method for processing disfluencies and out-of-vocabulary words in the context of automatic Arabic speech understanding. Our method based on a robust and partial analysis of Arabic oral utterances (conceptual segments analysis) is effective for the treatment of such phenomena. This method has been tested through the understanding module of SARF system, an interactive vocal server for Tunisian railway information.

1 Introduction

Spoken Arabic has been the subject of few researches compared to other languages such as English and French. There are at least two reasons for this, one is the lack of available speech corpora and another is due to the characteristics of Arabic speech. According to our knowledge, only one study has been done on automatic Arabic speech understanding by Zouaghi et al. (2008) as part of the Oreillodule project. However, no work has been done on the treatment of the two omnipresent phenomena in Arabic speech interaction namely, disfluencies due to the spontaneity of interaction and Out-Of-Vocabulary

(OOV) words due to the errors of speech recognition.

Thus, in this paper, we propose a method for disfluency processing (specifically, repetitions, self-corrections and word-fragments) and out-of-vocabulary words (particularly, misrecognized, unknown and truncated words) for automatic Arabic speech understanding. This method is based on a robust and partial analysis of oral Arabic utterances. Indeed, an utterance semantically labeled undergoes three levels of treatment: *i*) conceptual segments tokenization, *ii*) detection and correction of the disfluencies and *iii*) OOV word processing.

As an application, we have chosen the case of an interactive vocal server for information about Tunisian national railway company. The objective of this vocal server is to allow the user to communicate with the machine, via Modern Standard Arabic speech for Tunisian railway information (e.g., train schedule, fares, etc.).

In the context of this work, we are interested in Modern Standard Arabic for three reasons: *i*) it is understandable and used in all Arab countries *ii*) it is difficult for any understanding system to handle different dialects *iii*) the absence of tools for Arabic dialects.

2 What are Disfluencies and OOV Words?

Among the spoken language irregularities, known as disfluencies (Bove, 2008), we quote:

- **Repetitions:** are the identical series of words (or group of words) and the same syntactic category. A repetition may be

partial (part of a phrase) or total (full syntagm).

- **Omissions:** are caused by the absence of one or more syntactic constituents.
- **Self-corrections:** are the corrections made by the speaker himself to correct his utterance. This, then, is a correction of a single word with another word (or phrase) or a correction of an entire segment by another segment.
- **Restarts:** are disruptions followed by a new syntagm. Then there is abandonment of a segment and the beginning of a new segment.
- **Word-fragments:** are words started and unfinished. The result is then some fragments of words, as the case may be dropped or taken up and completed by the speaker.

Among the OOV words we quote:

- **Misrecognized word:** is a word that is produced in the output of speech recognition, while another word was pronounced.
- **Unknown word:** is a nonexistent word in the lexicon of the module of the recognition or the understanding.
- **Truncated word:** is a word recognized in part by the speech recognition.

3 Related Works

In this section, we briefly outline the main works on the disfluency and OOV word processing.

3.1 Disfluency Processing

For disfluency processing, we distinguish three main approaches: the Stanford Research Institution (SRI) approach, the stochastic approach and the linguistic approach.

- **SRI approach:** This approach is one of the early works on the disfluencies. The first step of this approach proposes a scheme for annotating disfluencies (Bear et al., 1992). This scheme combines simplicity and fitness necessary for the representation of different forms of disfluencies. This approach combines syntactic and semantic analysis (to reduce

the increasing number of patterns) with the technique of pattern matching (to detect and correct simple repetitions and simple syntactic errors as: "a the", etc.)

- **Stochastic approach:** This approach is based on the patterns. It is developed within the University of Rochester (Heeman and Allen, 1996). The first step of this approach proposes a modified version of the annotation scheme of the SRI approach. Thus, the proposed scheme does not allow the sharing of the area replaced in the case of complex disfluencies. To detect and correct disfluencies, this approach uses a language model combining different sources of information (the identity of words, syntactic information, transitions between words and the prosodic and acoustic index).
- **Linguistic approach:** This approach is developed by the dialogue group at the University of Rochester (Core and Schubert, 1999). In this approach, the processing is done in two steps: *i*) detection of disfluency boundaries using a statistical language model and *ii*) a syntactic analysis using meta-rules taking into account the relations between syntactic structures that dominate the words.

3.2 OOV Word Processing

Some research works have focused on the OOV word processing in the speech recognition. Among them we cite the work of Bazzi et al. (2001) limited to the treatment of names of cities considered as OOV words, and the work of Schaaf (2001) on the family names. The classical idea is to add to the basic model of the speech recognition module, an acoustic model of OOV words or to use the language model for OOV word processing.

Other research works have focused on the treatment of this phenomenon in the understanding speech. Among them we cite the work of Hazen et al. (2002) restricted to the treatment of the names of cities and the useless words, and the work of Bousquet-Vernhettes

(2002) on names of cities. The main idea is to detect and interpret the OOV words using the indications given by the rate of trust obtained in speech recognition.

4 Our Method of Disfluency and OOV Word Processing

According to the statistics obtained from our study corpus (see section 5.1), 25.24% of the utterances include disfluencies and 38.22% include OOV words. Both of these high percentages led us to propose a method of treatment of these two omnipresent phenomena in the Arabic utterances automatically transcribed.

$$\begin{array}{cccc}
 \langle \text{Ticket_Type, ذهاب} \rangle & \langle \text{Ticket_Type_Mark, تذكرة} \rangle & \langle \text{Tariff_Ref_Word, ثمن} \rangle & \langle \text{Request, أريد} \rangle \\
 \langle \text{Arrival_City_Mark, إلى} \rangle \langle \text{HV, سقف} \rangle \langle \text{Depart_City_Mark, من} \rangle \langle \text{Ticket_Type, إياب - ذهاب} \rangle \langle \text{Rectification_Mark, لا} \rangle & & & \\
 \langle \text{Minute_Mark, دقيقة} \rangle & \langle \text{Number, 33} \rangle & \langle \text{Number, 11} \rangle & \langle \text{Hour_Mark, ساعة} \rangle \langle \text{Months, مارس} \rangle
 \end{array} \quad (1)$$

4.1 Conceptual Segment Tokenization

This step uses conceptual segments consisting of classes of words. Indeed, a conceptual segment is a word sequence corresponding to the basic units of meaning (Bousquet-Vernhettes, 2002). Thus, a sequence of words making a conceptual segment is a segment of this concept. For example, the sequence of words من تونس [mn twns] (from Tunis) is a conceptual segment of *Departure*. We distinguish three kinds of conceptual segments: the *illocutionary* referring to the speech act theory (i.e., *Fare_Request*, *Dialogue_Start*, etc.), *Referential* for representing the domain of application (i.e., *Departure_Time*, *Departure*, *Destination*, etc.) and *Filler* regroups all words or word sequences judged as irrelevant for the meaning representation (i.e., *Noise*, *Digression*, etc.). For illocutionary conceptual segments we introduced a new conceptual segment that we named *Disfluent* containing the disfluencies. This

Recall that for disfluencies, our method is focused on solving complex repetitions, self-corrections and word-fragments. By complex repetitions, we mean the repetitions of two or more segments of words separated or not by words marking hesitation. For OOV words, our method allows the treatment of misrecognized words, unknown words and truncated words.

Thus, an utterance semantically tagged undergoes a conceptual segment tokenization, disfluency processing and OOV word processing. To explain each step of our method, we propose the example (1) that represents an Arabic utterance semantically tagged.

segment will be an object of downstream stage processing.

At this stage of analysis, refinement of the semantic tags is mandatory for limiting the conceptual segments. This refinement is mainly based on the tag and the position of word in the utterance. Indeed, our method takes into account the context of the word in the utterance. Thus, the *Number* tag can have several possible refinements (e.g., *Fare*, *Hour*, *Minute*, etc.) depending on the context of the word which had this tag.

Thus, any utterance can be segmented into a series of conceptual segments as illustrated by example (2). This segmentation is based on the list of conceptual segments, pre-markers and post-markers in the utterance and the semantic tags of the words of the utterance.

Thus, the statement (2) is the result of the conceptual segments cutting of utterance (1).

$$\left\{ \begin{array}{l} \text{من سقف} \\ [mn sqf] \\ \text{(from roof)} \end{array} \middle| \text{Departure} \right\} \left\{ \begin{array}{l} \text{تذكرة ذهاب أه - لا ذهاب - إياب} \\ [t * krp * hAb |h - lA * hAb - Iy~Ab] \\ \text{(single ticket euh - no return ticket)} \end{array} \middle| \text{Disfluent} \right\} \left\{ \begin{array}{l} \text{أريد ثمن} \\ [OrAd vmn] \\ \text{(to want fare)} \end{array} \middle| \text{Fare_Request} \right\} \quad (2)$$

$$\left\{ \begin{array}{l} \text{33 دقيقة} \\ [33 dqyqp] \\ \text{(33 minutes)} \end{array} \middle| \text{Departure_Minute} \right\} \left\{ \begin{array}{l} \text{على - ساعة 11} \\ [Ely - sAEp 11] \\ \text{(at - hour 11)} \end{array} \middle| \text{Departure_Hour} \right\} \left\{ \begin{array}{l} \text{إلى مارس} \\ [Ily mArs] \\ \text{(to March)} \end{array} \middle| \text{Destination} \right\}$$

4.2 Disfluency Processing

The disfluency processing is to correct the disfluent conceptual segments detected in the

utterance tokenization phase. For this, the disfluent segment undergoes an annotation similar to that proposed by Bear et al., (1992), and then it is corrected. The segment is described as a

succession of *Reparandum* (part of the segment to be corrected later), an optional *Editing Term* (marker recovery) and *Alteration* (part that

$$\left[\begin{array}{c} \text{ذهاب - إياب} \\ [* hAb - Iy \sim Ab] \\ \text{(return)} \end{array} \right]_{\text{Alteration}} \left[\begin{array}{c} \text{أه - لا} \\ [[h - lA] \\ \text{(euh - no)} \end{array} \right]_{\text{Edition Term}} \left[\begin{array}{c} \text{ذهاب} \\ [* hAb] \\ \text{(single)} \end{array} \right]_{\text{Reparandum}} \begin{array}{c} \text{تذكرة} \\ [t * krp] \\ \text{(ticket)} \end{array} \quad (3)$$

At this level of analysis, the patterns of shallow detection of disfluencies are applied. They concern the case of a repetition or a self-correction. These patterns are based on the identification of sequences of words reparandum and alteration that are repeated in the same way (*M*), which are used (different words playing the same syntactic or semantic role: *R*) or that are added (Neutral words: X). There is also possibly an editing term (*ET*) and a point of interruption noted by a vertical bar (|). For example the pattern¹ "*RI ET | RI*" will be applied on the utterance (3), where the first *RI* (i.e., in the right) corresponds to ذهاب [*hAb] (single), the *ET* corresponds to أه-لا [[h-lA] (euh-no) and the second *RI* (i.e., in the left) corresponds to ذهاب-إياب [*hAb-Iy~Ab] (return).

$$\begin{array}{c} \text{تذكرة ذهاب} \quad \text{أه - لا} \quad \text{ذهاب - إياب} \\ \text{R1} \quad \text{ET} \quad | \quad \text{R1} \end{array} \quad (4)$$

For the correction itself, the alteration is kept, however, the editing term and the reparandum are deleted. The result segment, suffers a similar analysis to that of the phase of conceptual segments tokenization to determine the type of the result segment.

The segment (5) represents the disfluent segment of utterance (2) after correction:

$$\left\{ \begin{array}{c} \text{تذكرة ذهاب - إياب} \\ [t * krp * hAb - Iy \sim Ab] \\ \text{(return ticket)} \end{array} \right\} \text{Ticket} \quad (5)$$

4.3 OOV Word Processing

The OOV word processing is to detect and correct these words. Recall that an OOV word can be an unknown word, a misrecognized word or a truncated word. The detection of misrecognized words is more difficult than that of unknown words and truncated words, seeing that they are

corrects or completes the reparandum). For example:

detected in the morpho-logical analysis during pretreatment of the utterance. Indeed, the difficulty of judging that a word is misrecognized resides in this latter's belonging in the lexicon despite its confused with another word that was really pronounced. The aim is to assign to such words *HV* ("Hors-Vocabulaire") tags in order not to be interpreted as such before the correction.

After the utterance tokenization into conceptual segments, each segment word is matched with the appropriate conceptual segment word class. In case of matching failure, the segment word is considered as misrecognized and is tagged as an HV word. Consider the conceptual segment *Destination* إلى مارس [Ily mArs] (to March) of utterance (1). In this example, the word مارس [mArs] (March) although it is in the lexicon, is an OOV word to the class containing the names of cities (the information awaited is an arrival city not a month). So the word has been misrecognized and a HV tag will be awarded. This indication on the nature of the expected information allows the detection of misrecognized words and corrects them.

The correction of OOV words is, first, meant to assign the correct class where they normally belong to. Then we search in the identified class the nearest word to the OOV word. For example, after the detection of the misrecognized word سقف [sqf] (roof), it is assigned to the *City* class, as the desired information is a city. For the search of the word closest to سقف [sqf] (roof), a Levenshtein distance is calculated between this word and each word of the city class. Levenshtein distance *d* between two words is defined as the minimum number of editing operations (insertion, omission or substitution of a character) needed to transform a word into another. The word used is the one that had the smallest distance *d* less than or equal to threshold acceptance *S*, which we defined as follows:

$$s = \frac{\text{Number of characters of word1} + \text{Number of characters of word2}}{2} \quad (6)$$

¹ The patterns are read from right to left because they are specific to the Arabic language.

In the case of failure of the correction (i.e., no word is accepted or more words are allowed), the search is redone with relief Levenshtein algorithm. On the assumption that long Arabic vowels (i.e., $[A]$, $[w]$ and $[y]$) can be inserted or omitted by the speech recognition and characters phonetically close, can be substituted by one another (e.g., « $[s]$ –ص– $[S]$ », « $[t]$ –ث– $[v]$ », « $[d]$ –د– $[*]$ », etc), another distance of Levenshtein d' is calculated by ignoring these

$$\left\{ \begin{array}{l} \text{من صفاقس} \\ [mn SfAqf] \\ \text{(from Sfax)} \end{array} \middle| \begin{array}{l} \text{Departure} \end{array} \right\} \left\{ \begin{array}{l} \text{تذكرة ذهاب – إياب} \\ [t * krp * hAb - Iy \sim Ab] \\ \text{(return ticket)} \end{array} \middle| \begin{array}{l} \text{Ticket} \end{array} \right\} \left\{ \begin{array}{l} \text{أراد ثمن} \\ [OrAd vmn] \\ \text{(to want fare)} \end{array} \middle| \begin{array}{l} \text{Fare_Request} \end{array} \right\} \quad (7)$$

$$\left\{ \begin{array}{l} 33 \text{ دقيقة} \\ [33 dqyqp] \\ \text{(33 minutes)} \end{array} \middle| \begin{array}{l} \text{Departure_Minutes} \end{array} \right\} \left\{ \begin{array}{l} \text{على – ساعة 11} \\ [Ely - sAEp 11] \\ \text{(at - hour 11)} \end{array} \middle| \begin{array}{l} \text{Departure_Hour} \end{array} \right\} \left\{ \begin{array}{l} \text{إلى قابس} \\ [Ily qAbs] \\ \text{(to Gabes)} \end{array} \middle| \begin{array}{l} \text{Destination} \end{array} \right\}$$

5 Presentation of SARF System

In this section we present our SARF system (“*Serveur vocal Arabe des Renseignements sur le transport Ferroviaire*”). SARF is an interactive Arabic vocal server that offers users access in oral modern standard Arabic to Tunisian railway information. It is based on the frame grammar formalism (Bruce, 1975) for oral utterance understanding and a selective approach. In what follows, we present the study corpus that we used to determine the semantic frames of our grammar, the Arabic lexicon relevant to railway domain, conceptual segments and the patterns of shallow detection of the disfluencies.

5.1 Study Corpus

Having an accurate and in-domain study corpus helps tremendously when creating a usable and accurate dialogue system. This is because a developer can accurately predict what vocabulary

editing operations. And if the problem persists, the word in question is supposed to be an OOV and keeps the HV tag.

In the utterance (2), the two words سقف [sqf] (roof) and مارس [mArs] (March) will be replaced by the words صفاقس [SfAqs] (Sfax) and قابس [qAbs] (Gabes). Thus, the utterance (2) becomes the utterance (7) after the disfluency and OOV word processing.

is needed and how the user’s input is structured based on the real-world examples in the corpus. However, as we noted at the beginning of this paper, the Arabic language resources are very rare and nearly unavailable. This is the case of Arabic speech corpora. Thus, within our application, we were obliged to build our own study corpus using the technique of *Wizard of Oz*.

Thus, we have used scenarios dealing with information on Tunisian railways. All queries were recorded and then manually transcribed according to standards of transcription in XML files, and tagged in accordance with the standards proposed by the ARPA community (Minker and Bennacef, 2005). We distinguish three types of queries namely, context independent queries (type A), context dependent queries (type D) and aberrant queries (type X). The following table summarizes the statistics on our study corpus.

Number of users	Numbre of dialogues	Size in hours	Number of words	Number of utterances			
				Type A	Type D	Type X	Total
50	300	11	92598	3356	4015	219	7590

Table 1. Characteristics of Our Study Corpus.

We have automatically transcribed the study corpus to study disfluencies and OOV words. Figure 1 shows the results obtained. Thus we have obtained a large number of disfluent utterances (25.24% of utterances) and utterances containing

OOV words (38.22% of utterances). These results justify the interest to consider the treatment of these two phenomena (i.e., disfluencies and OOV words) in the automatic spontaneous Arabic speech understanding.

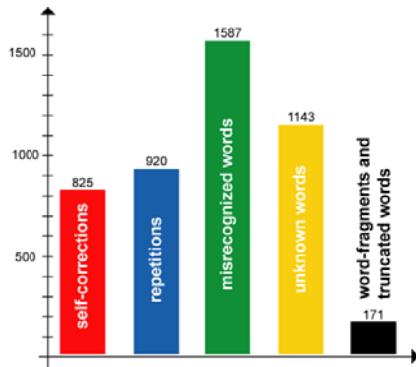


Figure. 1. Statistical Study on the Disfluencies and OOV Words in Our Study Corpus.

The study of this corpus has allowed us to identify six concepts namely, *Travel_Tariff*, *Travel_Schedule*, *Journey_Time*, *Ticket_Reservation*, *Train_Itinerary* and *Train_Type*.

A semantic frame is associated with every concept. Such semantic frame contains reference words and semantic cases related to our

application domain. We grouped the words that are semantically related in sets, and then we have assigned to each semantic frame the sets referring to it. In order to reduce the number of reference words, we have kept in the semantic frames one reference word for each set characterizing this frame. Also, from this corpus, we have built the lexicon relative to our application domain. Furthermore, to reduce the size of this lexicon, each word is reduced to its canonical form (stem).

5.2 Architecture of SARF System

The SARF system is composed of four modules namely, the speech recognition module, the speech understanding module, the dialogue management module and the speech synthesis module. Figure 2 shows the general architecture of our SARF system.

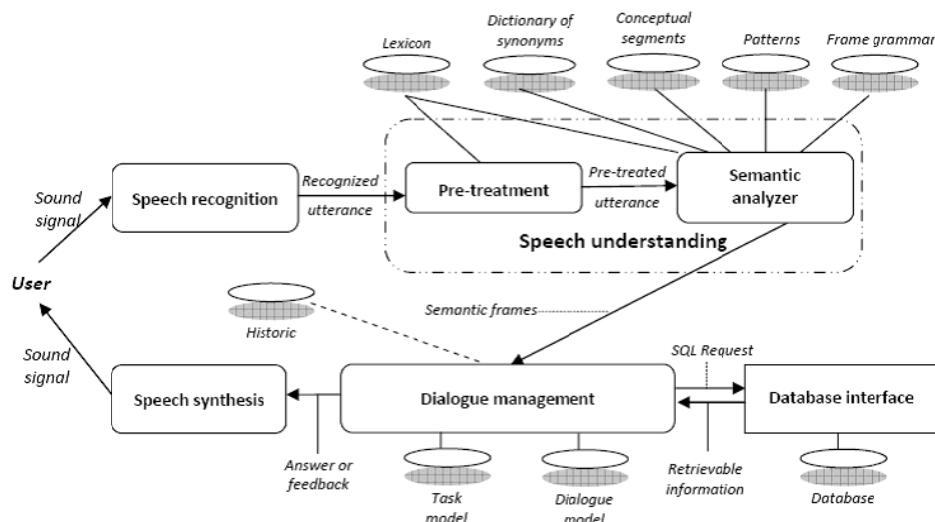


Figure. 2. The General Architecture of SARF System.

SARF integrates the techniques of speech recognition, understanding, dialogue management and speech synthesis. These techniques make possible the extraction of the meaning of an utterance pronounced by the user, in order to provide her or him with the required information. From the signal emitted by the speaker, the speech recognition module generates one or several lists of words that are supposed to correspond to the source utterance. The understanding module provides for the dialogue manager one or more semantic representations of

the transcribed utterance. The dialogue manager assures the interface with the database and suggests answers, or demands additional information to the user. The speech synthesis module allows their transmission into sound signal.

In what follows, we focus on SARF understanding module. Note that for the speech recognition module and the speech synthesis module, we plan to use commercialized systems. The understanding module is made up of two sub-modules namely, *i*) the pretreatment which

assures the normalization and the morpho-logical analysis of oral utterances and *ii*) the semantic analyzer which allows the utterance semantic tagging, the disfluency and OOV word processing and the semantic frames generating.

These semantic frames are stored in *XML* files before being forwarded to the dialogue management module.

The SARF understanding module is implemented with the *JBuilder 2007* environment using the *JAVA* programming language.

6 Evaluation of SARF Understanding Module

We built our evaluation corpus using the same technique of the Wizard of Oz used to build the study corpus. The evaluation corpus consists of 2823 requests (34726 words) of different types (1291 utterances of type A, 1434 utterances of type D and 98 utterances of type X), pronounced in a spontaneous way and automatically transcribed. The following figure shows the statistics on the used evaluation corpus.

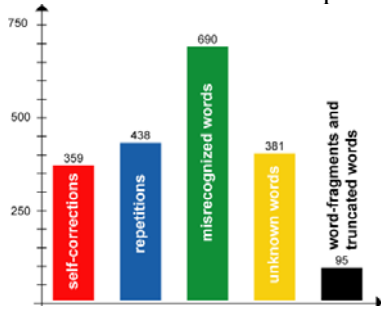


Figure 3. Statistical Study on the Disfluencies and OOV Words in the Evaluation Corpus.

To validate the method that we have proposed in this paper, we have evaluated our system on utterances containing disfluencies and OOV words. Thus obtained results are shown in table 2. Thus, we note that our system allows the treatment of a considerable number of disfluencies and OOV words. This robustness has reduced the failure cases of the semantic frame filling. Indeed, the failure cases obtained at the first evaluation without taking into account the treatment of these two phenomena was 18.54%. However, these failure cases are reduced to 12.63% when handling the phenomena of disfluencies and OOV words (i.e., which means a

reduction of 5.91%). After the evaluation of SARF understanding module, the measures of recall, precision and F-Measure that we have obtained are respectively 79.23%, 74.09% and 76.57% and the average time of execution of an utterance of 12 words, is at approximately 0.394 seconds.

	OOV words		Disfluencies	
	Number	Percentage	Number	Percentage
Correctly detected	910	78.04 %	486	61.29 %
Well corrected	795	87.36 %	394	81.07 %

Table 2. The Evaluation Results of SARF System.

The failure cases were mainly explained either by not detecting disfluencies or OOV words or by their non correction even though they are well detected.

For OOV words, their failure detection is mainly due to the wrong segmentation of the utterances into conceptual segments. This causes an assignment failure of those words to corresponding classes. In example (8), the misrecognized word ذهاب [*kɾp] (single) (The correct word is شباب [ʃbAb] (youth)) is assigned with error to *Ticket_Type* class and not to *Ticket_Category* class, generating a wrong tokenization, and then the non correction of this word.

$$\left\{ \begin{array}{l} \text{تذكرة ذهاب} \\ [t * kɾp * hAb] \text{Ticket} \\ (\text{single ticket}) \end{array} \right\} \quad (8)$$

Note that some OOV words that are correctly identified have several solutions. And thus, they can not be corrected. For example, the OOV word نابغة [nAbgp] (genius) is confused with the two words سابعة [sAbEp] (seventh) and رابعة [rAbEp] (fourth) of *Number* class ($d=d'=2$). Note that this ambiguity can be removed by the dialogue manager.

The failure of disfluency detection is mainly due to wrong tokenization of the utterances into conceptual segments. Especially in cases where the rectification markers are not included in the utterance. Example (9) shows the self-correction of the departure city صفاقس [ʃfAqs] (Sfax) by سوسة

[*swsp*] (Sousse), and seeing that the tokenization has produced three segments instead of one *Disfluent* segment, the system could not detect the self-correction seeing that there is not a *Disfluent*

$$\left\{ \begin{array}{l} \text{من سوسة} \\ [mn swsp] \\ \text{(from Sousse)} \end{array} \middle| \text{Departure} \right\} \left\{ \begin{array}{l} \text{إلى تونس} \\ [lly twns] \\ \text{(to Tunis)} \end{array} \middle| \text{Destination} \right\} \left\{ \begin{array}{l} \text{من صفاقس} \\ [mn SfAqs] \\ \text{(from Sfax)} \end{array} \middle| \text{Departure} \right\} \quad (9)$$

For the disfluencies, the cases that are well detected but not corrected are mainly the cases of complex disfluencies (presence in the same *Disfluent* segment of repetitions and self-corrections). Note that such ambiguities can be removed by the enrichment of the used patterns.

7 Conclusion and Perspectives

In this paper, we proposed a method for the disfluency and the OOV word processing in Arabic speech understanding. This method is based on three main stages namely, the segmentaion of the Arabic utterance into conceptual segments, the disfluency processing and the OOV word processing.

This proposed method was tested through the understanding module of SARF system, an interactive vocal server offering users access to Tunisian railway information. The evaluation of this module showed a decrease in error rate by 5.91%. These results are encouraging even if the module understanding of SARF in its current version does not treats certain phenomena such as *i*) complex disfluencies, *ii*) the complex self-correction where the corrected segment is farther than the wrong segment without involving rectification markers, *iii*) and the resumed in the form of abandonment of a segment to start a new segment.

As perspectives, we plan to study the types of untreated disfluencies to provide solutions for their detection and resolution. We also plan to use the dialogue manager to ask questions to the user in order to remove ambiguities.

References

Bazzi I. and Glass J. 2001. Learning Units for Domain-Independent Out-Of-Vocabulary Words Modeling. European Conference on Speech Communication and Technology, pp. 61--64. Aalborg, Denmark.

segment. Note that, for the example (9), our understanding module did not generate an error in filling semantic frame because it retains the last value of semantic cases.

Bear J., Dowding J. and Shriberg E. 1992. Integrating Multiple Knowledge Sources for Detection and Correction of Repairs in Human-Computer Dialog. In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, Newark, Delaware, USA.

Bousquet-Vernhettes C. 2002. Compréhension Robuste de la Parole Spontanée dans le Dialogue Oral Homme-Machine – Décodage Conceptuel Stochastique. PHD Thesis, University of Toulouse III–Paul SABATIER, France.

Bove R. 2008. A Tagged Corpus-Based Study for Repeats and Self-Repairs Detection in French Transcribed Speech. Proceedings of the 11th International Conference on Text, Speech and Dialogue, pp. 269--276. Brno, Czech Republic.

Bruce B. 1975. Cases Systems for Natural Languages. Artificial Intelligence, vol. 6, pp. 327—360.

Core M. and Schubert L. 1999. A Model of Speech Repairs and Other Disruptions. Presented at AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems, Cape Cod, MA, USA.

Hazen T.J., Seneff S. and Polifroni J. 2002. Recognition Confidence Scoring and Its Use in Speech Understanding Systems. Computer Speech and Language, pp. 49--67. Academic Press, vol. 16, N° 1.

Heeman P.A. and Allen J.F. 1996. Combining the Detection and Correction of Speech Repairs. In Proceedings of International Conference of Spoken Language Processing, pp. 363--365. Philadelphia, PA, USA.

Minker W. and Bennacef S. 2005. Speech and Human-Machine Dialog. Computational Linguistics. Vol. 31, pp. 157—158.

Schaaf T. 2001. Detection of OOV Words Using Generalized Word Models and a Semantic Class Language Model. European Conference on Speech Communication and Technology, pp. 2581—2584. Aalborg, Denmark.

Zouaghi A., Zrigui M. and Antoniadis, G. 2008. Compréhension Automatique de la Parole Arabe Spontanée : Une Modélisation Numérique. TAL, vol. 49, N°1, pp. 141—166.