A Generic Tool for Creating and using Multilingual Phrasebooks

Michael Zock[1] and Guy Lapalme[2]

[1]Laboratoire d'Informatique Fondamentale (LIF), CNRS, UMR 6166
Case 901 - 163 Avenue de Luminy, F-13288 Marseille Cedex 9, Marseille, France

[2]RALI-DIRO, Université de Montréal
C.P. 6128, Succ. Centre-Ville, Montréal, Québec, Canada

Abstract. To speak fluently is a complex skill. If reaching this goal in one's mother tongue is already quite a feat, to do so in a foreign language can be overwhelming. One way to overcome the *expression problem* when going abroad is to use a dictionary or a phrasebook. While neither of them ensures fluency, both of them are useful translation tools. Yet, neither can teach you to speak. We will show here how this can be done in the case of a phrasebook. Being interested in the learning of foreign languages, we have started to build a multilingual phrasebook (English, French, Japanese, Chinese) whose sentence elements, typically words or expressions, are clickable items. This fairly simple feature allows extending considerably the potential of the resource. Rather than learning merely a list of concrete instances (sentences), the user may learn in addition the underlying principles (patterns), that is, the generative mechanism capable to produce quickly analogous, but more or less different sentences. A similar feature may be used to extend the resource, by mining a corpus for sentences built according to the same principle, i.e. based on the same pattern, but this is work for the future. Two of the main goals of this paper are to present a method helping learners to acquire the skill of speaking (patterns augmented with rules), and to allow experts (teachers) to add information either to extend the database or to add a new language. We've started from English and Japanese, adding very quickly French and Chinese.

1 Introduction

No doubt, speaking a language is a difficult task and doing so in another one is even more of a challenge as one has not only to juggle many constraints —(determine what to say, find the corresponding words, perform the required morphological adjustments, and continue to plan the next segment while articulating, i.e. externalizing, the result produced so far)— but also learn how to do this in a new language. This latter task requires not only the learning of a lot of new knowledge (words, rules), but also the acquisition of skills, i.e. 'know-how' or 'procedural knowledge' (Levelt, 1975; deKeyser, 2007a). The focus of this paper will be on learning, or, more precisely, achieving fluency in speaking a foreign language. The tool is generic and will be

Zock M. and Lapalme G.
A Generic Tool for Creating and Using Multilingual Phrasebooks.
DOI: 10.5220/0003027800790089
In *Proceedings of the 7th International Workshop on Natural Language Processing and Cognitive Science (ICEIS 2010)*, page
ISBN: 978-989-8425-13-3

illustrated via a phrasebook built for 4 languages (English, French, Japanese and Chinese) augmented with an exercise generator, called DrillTutor.

Our multilingual phrase book, an open, customizable, web-based exercise generator and study tool has been conceived for novices studying foreign languages. Our goal is to help the learner reach the level of fluency needed to express their basic needs via language (survival level): ask for information; answer a question; solve a concrete problem by using language, etc. In sum, we'd like to help someone to learn the basic stock of phrases and expressions that are generally taught in the classroom, or that are acquired via a ***phrase book***, the latter being structured by tasks a tourist is likely to perform: ask for information in public places, do shopping, etc. Yet, we would like to go beyond this and help the learner not only to learn literally a stock of phrases and words (instance-based generation), but also (or, more importantly) the underlying principles (structures) to build similar sentences. To reach this kind of open ended generativeness we propose an electronic version of a method called "pattern drill" (Chastaing, 1969). There are two good reasons for this:

- in order to be able to perform ***automatically***, that is, without having to think about them, a whole set of tasks, we must **exercise** them, as otherwise we will forget or be unable to integrate them into a well staged whole, a prerequisite for **fluency** (deKeyser, 2007);
- different people have **different needs**. This being so, we propose to build an **open system**, allowing the user to tailor the tool to make it fit his or her needs. The tool is meant to be accessible via the web, but it could also be used on a PDA or a mobile phone.

2 Theoretical Background and Motivation of our Approach

Since we deal with language production, or more precisely, the acquisition of the skill to become fluent in speaking, let's briefly review the major approaches and contrast them with our own. While several authors have proposed an architecture to account for language production (Garret, 1980; Fromkin, 1993, Bock, 1995), the two best known models come from researchers with different backgrounds and goals: language processed by the human *brain* vs. language processed by *machines*. The model elaborated by Levelt (1989), a psychologist, is an attempt to account for the process people go through when producing sentences in real-time. Hence, production is incremental and possibly error prone. The model developped by the NLP community (Reiter & Dale, 2000) is motivated by engineering considerations. The scope is text rather than sentences. Output is expected to be (immediately) perfect and the process is fully automatic. As one can see, while both models address apparently the same problem (language production), their goals, methods and scope are not the same at all. Let's take a quick look at each one of them.

2.1 Levelt’s « Speaker » Model

This model, based on empirical data like speech errors and disfluencies (Fromkin 1980; Garrett, 1991), is accepted by most psychologists. It has undergone various changes since its initial proposal (Levelt, 1993). From a bird eye’s view the process can be described as follows. The speaker starts to work out roughly the idea to convey (*conceptualization*). This preverbal message is mapped at the next stage on a corresponding linguistic form (*formulation*). The final step (*articulation*) consists in translating this metalinguistic form (noun, subject, verb, tense, etc.) into sounds, produced in the right order and with the proper intonation. To be a bit more precise, the components’ tasks are the following:

The **conceptualizer** decides what to say (message), at what level and in what tone. These choices take into account the speaker’s goals, the preceding discourse, the one still to come and inferable information.

The **formulator** takes care of the syntactic structure (ordering), lexical choices and morphological adjustments. Formulation involves two subprocesses: grammatical encoding and phonological encoding. The *grammatical encoder* accesses lemmas and syntactic information (part of speech). The task of the *phonological encoder* is to build a phonetic or articulatory plan for each lemma and for the utterance as a whole. To this end it inspects the lexical form which tells it about the morphology and the phonology of the unit under scrutiny, after which it will add information concerning syllables, hierarchical information like root and suffix, but also stress, etc. The result of this processing is a phonetic or articulatory plan, a sort of internal representation of how the planned utterance is meant to be articulated.

The **articulator** translates the received plan into movements, relying for this on his respiratory and muscular system to activate in a well-orchestrated way the appropriate organs (larynx, tongue…). Output is overt speech.

2.2 The Reiter and Dale Model

Reiter and Dale (2000) also decompose the process into three main stages:

Macroplanning comprises *content choice* and *document structuring*. The former decides what information to communicate explicitly in the text, given the interlocutors’ goals, knowledge, beliefs and interests, the latter deals with message clustering and message order to produce a thematically coherent whole without causing unwanted inferences (‘She got pregnant and they married.’ vs. ‘They married and she got pregnant.’). Cue words are possibly added to reveal or clarify the rhetorical role of the various conceptual fragments, i.e. clauses: ‘He arrived just in time (*despite / because of*) the heavy traffic’ (*concession* vs. *cause*).

Microplanning involves generation of referring expressions, lexicalization and aggregation. *Reference generation* consists in the description of the object referred to in such a way as to allow its distinction form a set of potential alternatives (the yellow car, the car, it). *Lexicalization* implies the replacement of concepts by words (DOG: canine, puppy), and finally *aggregation* consists in carving out the semantic space (graph representing the messages to convey) in such a way as to allow the integration of the conceptual fragments into a paragraph or sentence frame without making the structure too unbalanced. This step may imply elimination of redundant elements.

Realisation consists in converting abstract representations of sentences into concrete text, both at the language level (*Linguistic realization*) and the layout level (*Structure realization*): abstract text chunks (sections, paragraphs) being signaled via mark-up symbols.

2.3 Our Learning Model, a Hybrid Approach: Patterns, Rules or Both?

As mentioned already, and as it certainly has become obvious in the last two sections, producing language in real time is a complex process. We will present here below an approach, showing how the acquisition of this skill can nevertheless be made feasible. To see how it relates to the above work, we've tried to recast it in the Reiter & Dale framework presented above. Hence we will rely on terms like macro-level, micro-level, conceptual input, lexicalization, morphology, etc. However, before presenting our approach, we would like to stress the following point. Going through the steps just described and applying all the rules implied by the current generators and formal grammars is highly unrealistic for people trying to learn a foreign language. There are at least three reasons for making us doubt.

- *memory*: people don't have in their mind all the knowledge described, neither can they hold all the required information in their working memory (Baddeley, 1970);
- *attention*: people can focus only on a small set of items at a time;
- *time*: speech, i.e. the conception of a message and its translation into language, is extremely fast. Speakers don't have the time to do all the computations, i.e. search and apply the needed rules.

Linguists describe languages in terms of rules, but people hardly ever learn such descriptions, leave alone apply all of them, at least not at the initial stages of acquiring a new language. What people do learn though are patterns complying with these rules. Of course, people do use rules, but in conjunction with patterns. The latter can be seen as frozen instances of a given step in the derivational process.

Patterns can be abstracted at any level. They can be of any sort, hybrid, mixing semantic and syntactic information. Patterns are global structures, which can be built dynamically by applying a set of rules, but, they can also be stored as ready made sentence plans or templates in which case they behave somehow like words: they can be retrieved in one go, sparing us the trouble to have to go through the cumbersome process of structure creation. Obviously, access, i.e. pattern retrieval is much faster than its computation, i.e. the creation of given structure. There are simply too many steps involved. This is probably the reason why so many people use them for language (learners, interpreters, journalists, etc.) or other tasks (music, programming, chess playing, etc.) without even being aware of it (Nagao, 1984). Of course, there is a price to be paid : patterns need to be accessed (see below.) and they may need to be accommodated. In other words, patterns have qualities, but also some shortcomings:

they are rigid and tax memory. Imagine that you'd abstract a pattern for every morphological variation. Take for example the following two sentences and their respective patterns (I *go* to NY this summer [`I go <place> <time>`] vs. I *went* to London last week [`I went <place> <time>`]). It clearly doesn't make sense here to abstract two patterns, since the two sentences are so much alike. It would be much more reasonable to have one general pattern for the global structure (person go <place><time>) and a set parameters, i.e. rules for local adjustments, like agreement, tense, etc.

Just like patterns, rules have certain shortcomings. While they may account for the expressive power and all the regularities of a given language, they may prevent us from getting the job done in time, in particular if there are too many of them. This being so, we suggest to use a mixed approach, resorting to each strategy when they are at their best, patterns for *global structures*, the syntactic layout, i.e. sentence frame, and rules for *local adjustments*. This combination gives us the best of both worlds, minimizing the use of computational resources (attention, memory), while maximizing the power (speed) and flexibility of output (possibly needed accommodations).

When people learn a new language, they build some kind of database composed of words, patterns and phrases. This memory can consist of translation pairs, or, pairs of conceptual patterns and corresponding linguistic forms (sentences). One can also think of conceptual patterns as a pivot, mediating between translations of languages.

One problem with databases is access. As the number of patterns grows, grows the problem of accessing them. This is where indexing plays a role. Patterns can be indexed from various points of view: semantically (thematically, i.e. by domain), via the words they contain, syntactically, etc. While we index our patterns pragmatically, i.e. in terms of communicative goals (function the pattern is to fulfill), we allow access also via other means.

To see how our model relates to the generation models presented here above, we've tried as far as possible to recast it in those terms. The resource can be used both as a translation aid, as an exercise generator (our concern in this paper), or as a tool to extend the current database (this is work for the future).

In the first case it would function in the following way: given some user input (sentence), the system tries to find the corresponding translation, which is trivial if the translation is stored in the DB.[1] In the second case, the assumption is that the user knows the goal s/he'd like to achieve. Hence, the dialogue goes as follows. Given some goal (step-1), the system presents a list of patterns, from which the user must choose (step-2), pattern whose variables he will instantiate with lexical items (step-3) and morphological values (step-4).

It should be noted, that in this case *conceptual input* is distributed over three layers: at a global level (macrolevel) the speaker chooses the pattern via a *goal*, by providing

[1] If the goal is the extension of the database by finding similar sentences in a corpus, i.e. sentences built on the same pattern, the problem is harder. The program must infer or abstract the input's underlying pattern, and find a corresponding sentence in the target language. This sentence can be either the translation of the input or a somehow similar sentence extracted automatically from the corpus. This is clearly work for the future. The main part of this paper deals with the *exercise generator*.

Table 1. Conceptual input as a four-step process.

task	input	output
MACROLEVEL	**1**) choice of *goal*	set of sentence frames 1 <OBJECT$_1$> is more <ATTRIBUT> than <OBJECT$_2$> 2 <OBJECT$_2$> is less <ATTRIBUT> than <OBJECT$_1$>
	2) choice of *sentence frame*	<OBJECT-$_1$> is more <ATTRIBUT> than <OBJECT$_2$>
MICROLEVEL	**3**) choice of *lexical value*	lexically specified structure *Perfume* is more *expensive* than *cigar*.
	4) morphological parameter	morphologically specified structure <OBJECT$_2$>: plural Fully specified concepual, syntactic and morphological structure

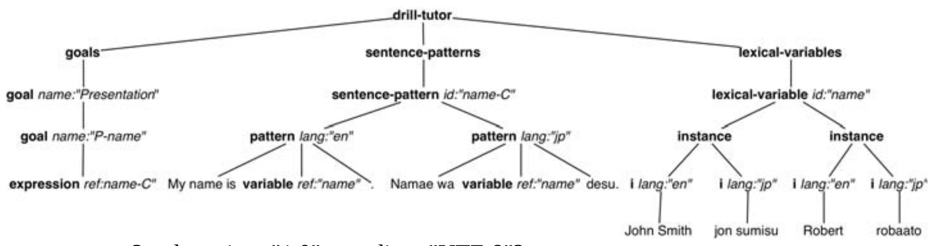
incrementally *lexical values* (for the pattern's variables) and *morphological parameters* (number, tense) to refine little by little the initially underspecified message. This kind of distribution has several advantages. Information is requested only when relevant and needed. There is better control in terms of access, storage and processing load. Obviously, this approach is better, than storing a pattern for every morphological variant. Last, but not least, this method is faster for conveying a message than navigating through a huge *lexical* or *conceptual ontology*, as suggested in (Zock,1991; Power et al., 1998; or Zock et al. 2009).

3 Building and using the Resource

Given the model described above, we want to clearly separate the description of the goals and the means, i.e. sentence patterns, to achieve them. Furthermore sentence patterns will be generalized in terms of variables standing for different words chosen according to a particular situation. For example, to achieve the goal of presenting oneself, one can use one of the following sentence patterns "They call me X" or "My name is X" in which the lexical variable X can be instantiated by the real name of the speaker. The allowed values of these variables can be defined in a dictionary. For a given goal, corresponding patterns and lexical variables can be defined for many languages.

Given the hierarchical nature of the goals and the flexibility needed for representing them, we decided to use an XML structure to keep the information. With an appropriate editor, it allows a linguist to define new goals, patterns and dictionary entries without dealing with the intricacies of the computer program that interprets this structure to display it to the user.

Figure 1 shows the tree and the corresponding XML structure for a simple pattern in two languages (English and Japanese).



```
<?xml version="1.0" encoding="UTF-8"?>
<drill-tutor>

 <goals>
  <goal name="Presentation">
    <goal name="P-name">
      <expression ref="name-C"/>
    </goal>
  </goal>
 </goals>


<sentence-patterns>
  <sentence-pattern id="name-C">
    <p lang="en">My name is <variable ref="name"/>.</p>
    <p lang="jp">Namae wa <variable ref="name"/> desu.</p>
  </sentence-pattern>
 </sentence-patterns>

 <lexical-variables>
  <lexical-variable id="name">
    <instance>
      <i lang="en">John Smith</i><i lang="jp">jon sumisu</i>
    </instance>
    <instance>
      <i lang="en">Robert</i><i lang="jp">robaato</i>
    </instance>
  </lexical-variable>
 </lexical-variables>
</drill-tutor>
```

Fig. 1. Definitions of *goals*, *sentence patterns* and *dictionary entries* in tree form and in the corresponding XML.

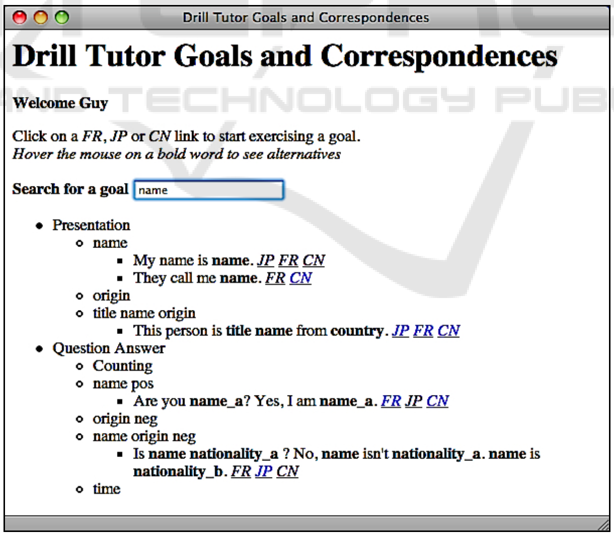
4 Example of Use

The user chooses the goal of communication. The goals being tree-structured, the user can either drill down to a given goal by clicking on a goal name to expand the sub-goals until a pattern is displayed with variables shown in bold. Goals being indexed it is also possible to search for them via a term. In Figure 2, the user asked for all goals and patterns dealing with the term "name". Having selected to exercise a given language, the user is shown the two steps displayed in Figure 3.

5 Technical Aspects and Current State

To implement the DrillTutor, we decided to use a web browser because it is readily available on all platforms and offers a powerful display mechanism capable to handle all commonly used languages and character sets provided one uses Unicode coded as UTF-8, which is the case for our XML files. To create these pages, we use PHP because it provides a convenient means for producing dynamic web pages and for accessing XML files. For dealing with Japanese, we used romaji, which are Latin characters that approximate the English pronunciation from which we developed a PHP function to compute the corresponding hiragana transcription for display in the browser.

Once a goal is selected, the PHP program generates a sentence from a pattern by finding values of each lexical variable and concatenating strings from the patterns with strings found for each variable. Variables being named can be used several times. As soon as a variable has been set, its value will be repeated across the respective patterns. Different variables for the same lexical value will get distinct values; this feature is used in patterns such as: *Are you Chinese? No I am not Chinese but Japanese* where *Chinese* and Japanese are considered as variables in the pattern. The first two occurrences would have the same name, but the last would be named differently.



Drill Tutor Goals and Correspondences

Drill Tutor Goals and Correspondences

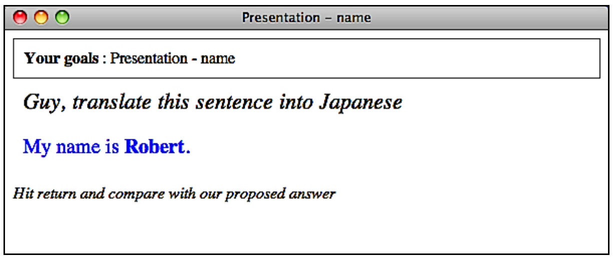
Welcome Guy

Click on a *FR*, *JP* or *CN* link to start exercising a goal.
Hover the mouse on a bold word to see alternatives

Search for a goal name

- Presentation
  - name
    - My name is **name**. JP FR CN
    - They call me **name**. FR CN
  - origin
  - title name origin
    - This person is **title name** from **country**. JP FR CN
- Question Answer
  - Counting
  - name pos
    - Are you **name_a**? Yes, I am **name_a**. FR JP CN
  - origin neg
  - name origin neg
    - Is **name nationality_a** ? No, **name** isn't **nationality_a**. **name** is **nationality_b**. FR JP CN
  - time

Fig. 2. Search and display all goals involving the *name of a person*. After each pattern are displayed links to exercise it in a given language.

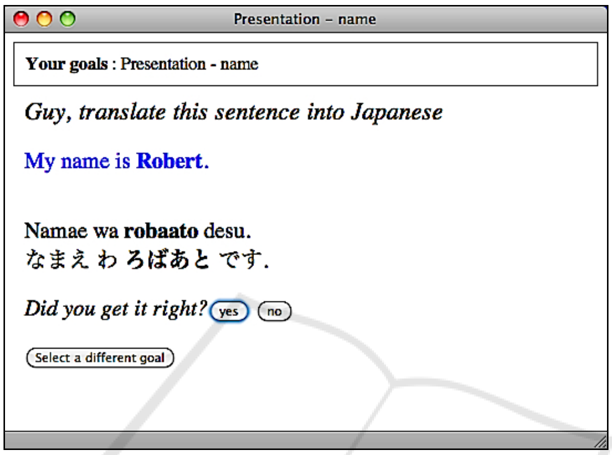

Presentation – name

Your goals : Presentation - name

Guy, translate this sentence into Japanese

My name is **Robert**.

Hit return and compare with our proposed answer


Presentation – name

Your goals : Presentation - name

Guy, translate this sentence into Japanese

My name is **Robert**.

Namae wa **robaato** desu.
なまえ わ **ろばあと** です.

Did you get it right?

Fig. 3. The top shows the initial screen with an instantiated sentence pattern. The user is invited to translate the prompted sentence in the chosen language and, after typing a carriage return; she is shown the proposed answer and asked if her answer was right. The number of correct or wrong answers to each goal is saved by the system and the user is shown another instance of the same pattern. It is also possible to choose another pattern (see Figure 2).

Statistics of the *performance* of each user are kept in a file on the server and its contents could be used to influence the selection of variables within a pattern and even the display of goals, depending on the perceived state of the learner. Currently though, variables within sentence patterns are selected randomly and all goals are displayed.

Javascript is used to allow a local interaction within a page. For example, in Figure 2, the web page initially contains all goals and patterns, but the display is controlled locally by Javascript code that handles mouse clicks and the content of the text field to display a menu of suggested keywords for finding goals.

The user interface of the system (i.e. the titles and the prompts) is localized using another XML file that stores the message strings in different languages (currently French and English).

The main motivation for separating the description of goals and patterns from the program was to enable a user to add new goals and patterns. This feature has not yet been implemented; hence the goals and patterns must be entered using an XML aware text editor. As we have defined a schema to validate the XML file, the editor can help in making sure that the information is entered in the right format and that all references to variables and patterns are existing elements in the XML file. Initially we had developed the patterns for English, French and Japanese. By editing the XML file, a collaborator managed to add Chinese patterns for all goals and the corresponding lexical variable entries within a few hours.

The current system has about 15 goals and patterns and 150 lexical items, in four languages with two interface languages. This looks quite small, but the reader should not get misled and bear in mind the following. The focus of our work was to check how difficult it would be to add a new language, possibly typologically very different, and as the result showed, even adding a language like Chinese, turned out to be quite simple. In sum, the number of patterns and the size of the vocabulary is not really what counts at this stage, the focus being on the implementation of an editor designed for building, modifying and using a database. The database can easily be extended. The architecture we have defined can handle as many goals, patterns and lexical items as wanted. All we need to do is to edit XML files by following a well-organized pattern (frame) whose well-formedness can be checked via a Schema-aware XML editor.

6 Discussion, Future Work and Conclusions

We have presented a method (multilingual phrasebook augmented with an exercise generator) whose main purpose is to help people acquire fluency in a new language. While the current coverage is still very small, we believe that the approach is sound, both for extending the resource and for assisting language learning (memorization of structures, i.e. words in context; acquisition of fluency). Concerning extensions we have tested the tool and could show that it was fairly easy to add new information, or even a typologically completely different language. We've argued that producing sentences by applying only rules was too cumbersome, and that patterns alone were too rigid and too greedy concerning storage (memorization) and access, hence we've settled for a compromise, using patterns for the global structure and rules for the local refinements. This improves speed considerably, while minimizing storage of patterns and rules.

Obviously, the ultimate judge will be the user, but since we are still in the initial phase of this project, this will have to wait for a while. Nevertheless, the first results look very promising, so does the feedback of our colleagues. The following features are characteristic of our enterprise: the system is *open* (new information can be added anytime), *generic* (other languages can be implemented very quickly), and the project *scales up*. Indeed, we plan to make the system self-extending, by mining a corpus for pattern instances, but this is still music for the future.

References

1. Baddeley, A. (1990). Human memory: theory and practice. Hillsdale: Erlbaum.
2. Bock, J.K. (1995). Sentence production: From mind to mouth. In J. L. Miller & P.D. Eimas (Ed.), Handbook of perception and cognition. Vol. 11: Speech, language and communication. Orlando, FL: Academic Press.

3. Chastain, K. (1969). The audio-lingual habit learning theory vs. the code-cognitif learning theory. IRAL, 7(2), 97–107.
4. deKeyser, R. (Ed.), (2007). Practicing in a second language: Perspectives from applied linguistics and cognitive psychology. Cambridge University Press
5. deKeyser, R. (2007a). Skill acquisition theory. In J. Williams & B. VanPatten (Eds.), Theories in Second Language Acquisition: An introduction (pp. 97-113). Mahwah, NJ: Erlbaum.
6. Fromkin, V. (1993). Speech Production. In Psycholinguistics. J. Berko Gleason & N. Bernstein Ratner, Eds. Fort Worth, TX: Harcourt, Brace, Jovanovich.
7. Fromkin, V. (Ed.), (1980). Errors in linguistic performance: Slips of the tongue, ear, pen and hand. New York: Academic Press.
8. Garrett, M. (1991). Sentence processing. In D, Osherson and H. Lasnik (Eds.), Language: An invitation to cognitive science, Cambridge, Mass.: The MIT Press.
9. Garrett, M. F. (1980). Levels of processing in sentence production. In B. Butterworth (ed.), Language production: Vol. 1. Speech and Talk, pp. 177-220. London: Academic Press.
10. Levelt, W. (1993) The architecture of normal spoken language use. In Blanken G., J. Dittmann, H. Grimm, J. Marshall & C. Wallesch (eds.) Linguistic Disorders and pathologies. W. de Gruyter, Berlin, New York
11. Levelt, W. (1989). Speaking. MIT Press, Cambridge, Mass.
12. Levelt, W.J.M., (1975). Systems, skills and language learning. In A. van Essen & J.P. Menting (Eds.), The Context of Foreign Language Learning (pp. 83-99). Assen: Van Gorcum.
13. Nagao, M. (1984). A framework of a mechanical translation between japanese and english by analogy principle. In A. Elithorn & R. Banerji (Eds.), Artificial and Human Intelligence (pp. 173–180). Amsterdam: Elsevier.
14. Power, R., Scott, D. & Evans, R. (1998). What you see is what you meant: direct knowledge editings with natural language feedback. In H. Prade (Ed.), 13th European conference on artificial intelligence (ECAI'98) (pp. 677–681). Chichester: Wiley.
15. Reiter, E., & Dale, R. (2000). Building natural language generation systems. Cambridge: Cambridge University Press.
16. Zock, M. (1991). Swim or sink: the problem of communicating thought. In M. Swartz & M. Yazdani (Eds.), Intelligent tutoring systems for foreign language learning (pp. 235–247). New York: Springer.
17. Zock, M., Sabatier, P. and L. Jakubiec. Message Composition Based on Concepts and Goals. International Journal of Speech Technology, 11(3-4):181–193, 2008.