Laboratorio de Tecnologías de Rehabilitación
Dpto. de Ingeniería Electrónica
Universidad Politécnica de Madrid

Contribution to word prediction in Spanish and its integration in technical aids for people with physical disabilities.

(Aportación a la predicción de palabras en castellano y su integración en sistemas de ayuda a personas con discapacidad física)

Author: Sira E. Palazuelos Cagigas

PhD. Thesis Advisor: Santiago Aguilera Navarro

Madrid, 2001

Index

Acknowledgements

Muchas son las personas a las que tengo que agradecer haber llegado hasta aquí. En primer lugar, a Santiago Aguilera, director de este trabajo, por haber apostado por mí, dándome la oportunidad de trabajar en este campo, que ha resultado ser apasionante, y por la enorme paciencia y el apoyo que me ha dado a lo largo de estos años.

Esta tesis me ha permitido conocer a gran cantidad de gente, que ha tenido mucha influencia en mi vida, tanto profesional como personalmente. El Laboratorio de Tecnología de Rehabilitación me ha proporcionado todo su apoyo profesional, además de conseguir convertir en agradables los largos días de trabajo. Gracias a todos los proyectandos Alberto, Luis, los Franciscos, Juan, Antonio, Tobar, Juanjo, Elena y David etc., por aportar su granito de arena.

Gracias a todos mis amigos y compañeros del Departamento de Ingeniería Electrónica de la ETSI de Telecomunicación de la UPM, donde se ha elaborado esta tesis, que han compartido conmigo estos años apoyándome.

Especialmente enriquecedoras han resultado mis estancias en el extranjero, en la Universidad de Sheffield; en la Universidad de Dundee, con Alan Newell, Ian Ricketts, Peter Gregor y todos los demás; en el KTH, en Estocolmo, donde disfruté de la calidad científica y humana de Sheri Hunnicutt, Hakan Melin y Alice Carlberger, entre otros; y en el GRIL de la Universidad de Clermont Ferrand, donde tuve tantas fructíferas discusiones con Gabriel Bes y José L. Rodrigo, que posteriormente fueron continuadas en España con Francisco Aliaga.

A lo largo de los años, he disfrutado de la comprensión del Departamento ICS de la EUIT de Telecomunicación de la UPM, y por supuesto, del Departamento de Electrónica de la Universidad de Alcalá, donde no ha pasado un sólo día sin que alguno de sus miembros tuviera para mí palabras (y muchas obras) de ánimo. Gracias a Manuel Mazo, José Luis Lázaro y todos los demás y, sobre todo, a los habitantes de los pasillos 21 y 22.

Esta tesis no habría sido la misma sin el CEAPAT, sus proyectos, sus profesionales (Cristina Rodríguez, Cristina Larraz, Lola Abril y tantos otros) y la valiosa información que ha puesto a mi disposición, que espero haya proporcionado los frutos deseados, aportando un granito de arena a la mejora de la calidad de vida de las personas con discapacidad. Muchas gracias también a Amparo Candelas y Manuel Lobato. También me gustaría agradecer todas las críticas, sugerencias, comentarios e informes recibidos, especialmente a David Carreres y Sergio, del Centro Maset de Frater, y a Montse, que me han ayudado a mejorar mi trabajo, a la vez que me animaban a seguir adelante.

En el plano personal, he tenido mucha gente cerca. Quiero dar las gracias a las familias Baena-Moreno y Romero-Marín, por haberme permitido compartir un trocito de sus vidas. A Javier Romañach, por sus estimulantes charlas sobre lo divino y lo humano. A Fernando González, por estar ahí cuando hace falta. A Pepe, porque su entusiasmo tiene parte de culpa de que yo esté aquí. A Javier Gamo, que aparece oportunamente en los momentos más insospechados de mi vida. A Jero, por los ratos de trabajo codo con codo. A Javi Rubio, Jorge, Cris y Olga, por las estupendas veladas que hemos disfrutado juntos. A Juan Ignacio (¡ánimo, ahora te toca a ti!), compañero de fatigas con quien he compartido tantos buenos y malos momentos.

A Javi y Mónica, por recordarme lo que significa la palabra amistad, y a Jose, porque siempre Está dispuesto a ofrecer una mano llena de pAz.

Por último, muchas gracias a toda mi familia y a Javi, por su cariño y apoyo incondicional y por soportar todas mis ausencias, que espero no se vuelvan a repetir.

Por supuesto, son todos los que están, pero no están todos los que son.

Muchísimas gracias a todos...

Abstract

The aim of this Ph.D. thesis is the study of the inclusion of linguistic information in word prediction for Spanish, being the main objective improving the writing aids available for people with different kind of disabilities.

In order to include linguistic information, a novel architecture that allows the development of an original methodology is proposed, so that the different sources of information that have been explored are combined (mainly in the lexical, morphological and syntactic levels). This combination is performed by a management module, able to deal with the different information flows used and combine them as well. Strict separation between the lexicons (main, custom and subject) and prediction methods is essential in the architecture and makes the combination possible.

The prediction methods included use two main modeling strategies for the linguistic information: stochastic modeling (unigrams, bigrams, bipos and tripos), and formal modeling (using a probabilistic context free grammar strengthened with additional characteristics). In every module including linguistic knowledge, specific contributions have been made, both in the design and organization of the information (mainly oriented to be used in the formal grammar) and also in the particular methodology of using this information when facing word prediction and the adequate cooperation with other modules. The design criterion and the definition of the grammatical parts-of-speech (*pos*) used are also considered to be significant contributions of this thesis. They are intended to better *connect* with the observed syntactic behavior, along with the design of a feature set towards which part of the expressive content has been shifted. In order to deal with both *pos* and features, some original mechanisms included in the design of the formal grammar are also proposed.

With respect to the formal model, the detailed study of linguistic phenomena (both theoretically and empirically) has led us to design a probabilistic context free grammar that uses an original interweaving of different mechanisms (terminal symbol feature concordance, imposition and prohibition; powerful feature management also in non terminal symbols; lemma and word imposition and prohibition; and the possibility of dealing with optional symbols) that endow it with a significant descriptive power of the language, while keeping the number of rules and the search process computationally tractable. This work is not only limited to a theoretical study. A working system has also been evaluated and implemented, built following the proposed architecture in which, additionally, specific considerations on the user interface design have been taken into. account

A detailed study on the different factors that affect the quantitative evaluation (where a normalization effort should be done, given the lack of defined standards on this topic) is also provided, proposing metrics able to analyze the power of the information sources that allows us to select the best combination strategy leading to actual improvements for the users of this technology. In this combination, we prioritize the words coming from the subject and custom lexicons using a bigram model. After this, the stochastic *pos* models is used, firstly applied to the subject lexicon and afterwards, with an adequate weighting, to the custom and main lexicons.

With respect to the word prediction method based in the formal grammar, the overall set of contributions allowed us to achieve results close to those obtained with the stochastic pos models, leaving for future research the completion of its descriptive capabilities. The modularity and flexibility of the architecture will allow us to carry out this research work taking great advantage of the effort already invested here.

1 Introduction

Word prediction is one of the most commonly used methods in communication aids for people with different sorts of disabilities, but able to read and write. The advantages of the word prediction depend on the degree of impairment of the user: for users with physical disabilities with good linguistic skills the system will only help them in the physical sense, accelerating the writing and reducing the effort needed to type. Users with linguistic problems (i.e., dyslexics, or people with problems to generate grammatically correct sentences) will write more correct phrases, if grammatical information is included in the prediction system, because they will be able to recognize the adequate words when shown in the menu.

Generally speaking, a word prediction application tries to find out which is the word a user is typing or is going to type, before he/she writes it completely. The guessed words are shown somehow, so that, if the desired one is included in that list, the user can select and insert it in the text, avoiding the need to type the rest of the word, thus reducing the time and effort necessary to write the text. This is especially useful for slow typists, or people who are not able to use a conventional keyboard. The methods to avoid its use usually involve the utilization of as many switches as the user is able to handle (usually 1 or 2) and a keyboard emulator, controlled with the switches. Because of the little versatility of the switch-based access, the use of matrices scanning is generally employed. As this is a very slow input method, several acceleration techniques are used to increase the user's typing rate. Word prediction is one of them.

People with other writing problems may also take advantages of the word prediction: dyslexic users or people with frequent spelling errors feel more confident writing with the help of this aid. Children and people learning a second language are also target users of these systems.

2 Background

In this section, the background of some important subjects that may be related with word prediction is outlined.

In the first section, some language models that may be used in word prediction are explained. The most simple models, based on words sequences, as well as the more powerful ones are described, adding some grammatical information in several complexity levels.

Afterwards, some considerations to be taken into account in the evaluation stage are presented. Finally, a set of commercial systems that use word prediction are shown.

2.1 Language models

In this section, some language models are described. Their advantages and disadvantages, the training and evaluating methods, and its possible use in the word prediction, the information they need, etc., are explained. The models described are:

- Statistical models based on statistics of words and sequences of words: unigrams, bigrams, etc., whose main problem is the great amount of text needed to train them. The negative effects of the lack of enough training text makes necessary to use techniques like matrices *smoothing* or grouping words in *parts of speech (pos clustering)*.
- *Statistical POS* models: instead of making statistics on words and their sequences, they are based on *pos* and their sequences, so that they do not need so much training texts. The problem is that this method requires more information: the set of *pos*, a tagged training text according to this set of pos and the grammatical information of every word (to be included in the dictionary). As the tagged text is difficult to obtain, because it requires a great effort (the tagging process is not fully automatic if you want to get the highest tagging accuracy), the amount of text usually available is only enough to train *bipos* or *tripos.*

  This information only models the relationships among two or three consecutive words, but there are long term relationships that need more powerful and complex models to be managed.
- Formal models: regular, context free, context sensitive and type 0 grammars have been studied. Context free grammars have been finally chosen as the base to predict, because they are powerful enough to describe the structures of a

somehow simplified natural language, and simple enough to build an efficient parser. Some other features can be added to these grammars: feature management (augmented grammars) and the possibility to deal with ambiguous structures in parallel, also adding statistical information.

Parsers are the processes that identify the structure of a sentence according to a given grammar. The sort of parsers needed to predict are left to right and top down parser, to be able to analyze the part of the sentence already written by the user, and offer grammatically correct predictions. After studying several different parsers, the Earley parser has been selected, because it can parse a context free grammar, it is very efficient, and it can be modified to include new features to increase its power.

One of the problems to solve in a parser is the error management. There are situations that stop the analysis of a sentence, like unknown words in the text or words that do not follow the rules (either because the rule is not included in the grammar, or because the sentence is not grammatically correct).

- Unification grammars.
- Grammars based on subcategorization frames.
- Left associative grammars.

2.2 Adaptive interfaces oriented to people with disabilities

The design of any program interface may follow known criteria that ensure: a consistent organization of the program menus, an optimum configuration (clear and simple, using the minimum resources to perform each task) and a user friendly configuration (requiring a small effort for the user to learn how to use the system).

It is strongly recommended to follow these criteria in the design of any interfaces, but it becomes especially important when dealing with interfaces for people with disabilities. The design of these interfaces is essentially conditioned by the needs of the users, who have problems to use the conventional input methods (keyboard, mouse, joystick, etc.). This also implies a redesign of the interface in order to be controlled only with the devices the user can handle (for example, if the user can only use a switch, the user interface should include scanning options). In Spain there are some initiatives to standardize the user interface design for people with disabilities, in the workgroup 1 of AENOR (Spanish Association for Standardization and Homologation) to which the author of this Thesis belong to.

In the design of the interface, as well as the solutions for each disability, other strategies, like the adaptability, can be considered, in order to accelerate and optimize the accesses to the program. The adaptability strategies included in conventional interfaces are, for example, the modification of the options location, automatic macros creation, and help agents to convey information depending on the user actions.

In the user interfaces for people with disabilities, the adaptability strategies should be modified, to adapt them to the users' particular characteristics. For example, one of the

changes proposed in this thesis: the automatic control of the scanning speed depending on the number of erroneous keystrokes.

2.3 Evaluation

Word prediction can be evaluated qualitative and quantitatively.

Quantitative evaluation consists of a study of the prediction results according to two criteria:

- Time and effort saved by the users, what becomes especially important for people with physical disabilities because of the theoretical improvement in writing speed.
- Number of predicted words, which is a very important aspect for people with linguistic problems (as dyslexia).

This automatic evaluation may be run for different configurations, allowing us to find out which is the optimum method or combination of methods. Of course, this is a theoretical optimum, and in a real system other considerations may be taken into account. For example the learning capabilities of subject lexicons lead to very good results in automatic tests, but they may be counterproductive, for instance, when a child who is learning to write uses them, because of the high number of misspelled words, that would be included in the dictionary and predicted, producing an undesired feedback.

Qualitative evaluation consists of making a detailed study of the text generated by the user, for example, comparing the quality of the text generated with and without word prediction. It is necessary to determine the criteria to measure the quality of the texts, for example the number of misspelled words, the intelligibility of the generated texts etc. A study of the effect of the word prediction in the text can be found in [Magn97a] and [Magn97b].

Apart from the quality of the text, the user's point of view is also important, the user's subjective opinion: if he/she likes the system, if it makes it easier for him/her to write, if he/she feels more confident using word prediction or the increase in the cognitive load makes the system difficult to use, etc.

2.4 Word prediction systems

In this section, a brief description of several commercial systems is presented: Aurora 3.0 (Aurora Systems), Co:Writer (Don Johnston, Inc.), EZ Keys (Words+, Inc.), PAL (Univ. Dundee), PredictAbility (Inclusive Technology Ltd.), Predice (LTR), Prophet (distrib. by ACE), SAW (Oxford ACE Centre), SoothSayer Word Prediction (Applied Human Factors), Telepatic (Madentec), TPV (distrib. By CEAPAT), Write Away 2000 (Information Services Inc.). Most of these systems predict only in English, although some of them are able to predict in more than one language, such as Swedish, Norwegian, Danish or Spanish.

3 Word prediction

As it has been shown in the previous chapter, there are several systems that include word prediction. Most of them use statistical methods to predict, being English the most common language.

In this PhD Thesis, a study of word prediction for Spanish has been done, considering the different information sources that could be used to increase its quality, and the method to include and combine the contribution of each technique. A system that includes all the explored methods has been built, making possible to verify the capabilities of each method, as well as to obtain the optimum configuration of the final prediction system.

In this chapter, the study followed to design the different prediction methods is described. The contributions made to each method are presented, as well as the information needed and the implementation methodology. The architecture of the whole prediction system is shown, highlighting the improvements made in each module (information source or prediction method).

3.1 General architecture

The general architecture of the prediction system is shown in Figure 1.

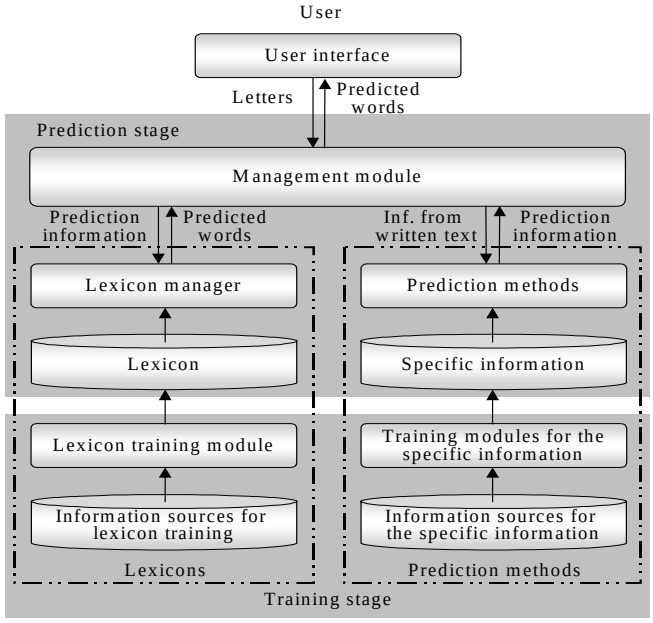

Figure 1. General architecture of the word prediction system.

As it can be seen, the prediction methods and the lexicons are independent, making the system design and the portability to other languages easier.

In the architecture the following main modules can be distinguished:

- *User interface*. It is the user interface of the program. It provides the management module with the user text, and shows the predicted words.
- Modules that take part in the prediction process:
  - *Management module*. It processes the input from the user interface (text written by the user, and manages the information flows between the different prediction methods (coordinating the data each one needs and provides) and the transactions with the dictionaries. It obtains the word prediction list that each method provides and sends the most adequate to the user interface.
  - *Lexicon managers*. They select the words that satisfy the constraints imposed by the prediction methods. They receive the constraints from the management module and return the list of words proposed by the corresponding lexicon.
  - *Lexicons*: General sources of information of the system. They contain the words and the probabilistic and grammatical information that the prediction methods need to select each word.
  - *Prediction method*s: They decide the constraints that the words actually satisfy, from the information provided by the management module and the specific information sources.
- Modules that take part in the training process:
  - *Lexicons training modules*: Which includes the procedures (automatic or not) needed to build the lexicons from the training sources. The information sources are texts, that may be tagged or not.
  - *Specific information training modules*: That involve the procedures (manual or automatic) needed to generate the specific information for each prediction method from the available training information sources: texts, tagged or not, and expert knowledge.

In the following sections the different modules of the system will be explained.

3.2 Lexicons

They are the basic knowledge sources for all the prediction methods and contain all the information of every word that the prediction methods need at any point of the prediction process.

From the study of the prediction system behavior, it was decided that different kinds of lexicons were needed. Each type of lexicon provides different information, and requires its particular training and management mechanisms. This division of the information in several lexicons can be found in other systems as Prophet or PredictAbility.

In this section, the need of each lexicon type is justified, as well as its contribution to the global system. The structure of each lexicon, the generation process, the management, and the possible use of each dictionary in the final prediction architecture are described.

3.2.1 Main or general lexicon

This lexicon contains the statistical and grammatical information needed for the different methods. It is static and it is used to obtain:

- Information about the words written by the user.
- The list of predicted words.

The main lexicon contains more than 160.000 entries and more than 130.000 different words, including words and punctuation marks. Each entry contains the word (the exact form), its lemma, its absolute frequency (in the training text), its grammatical category (part-of-speech, such as noun, verb, etc., according to the novel classification that will be presented in the section 3.2.1.1 page 16), and its features (for example, gender or number). Approximately a 20% of the lexicon words have homographs, with different grammatical information. In this case, there is a different entry for each one. As the semantic information is not considered, homographs with the same lexical information are represented by the same entry.

The generation of the main lexicon consisted of a series of automatic, semi-automatic and manual processes. The basic training text are extracted from the electronic version of the "El Mundo" newspaper (January to June, 1994). In the tagging stage we have used the automatic tagger SMORPH, developed in the *Groupe de Recherche dans les Industies de la Lange (GRIL, University of Blaise Pascal, Clermont Ferrand, France).* The main manual processes have been:

- The word list has been completed, adding missing words belonging to the *closed categories (*the ones that contain a small number of words, such as prepositions or determinants).
- An expert divided the absolute frequency of every word into its different homographs. Obviously, this is an approximation, because this information was not available, but it's accurate enough for our needs.
- The *pos* assigned to part of the words was modified following a new classification. A novel criterion has been added to the traditional ones, that clasifies the words according to their syntactic behavior: if two words have a different behavior, they are separated in different categories, even if they belong to the same category in traditional classifications.
- New features have been added to some words to augment their information. The set of features and its possible values are also different to the traditional ones for the same reasons.

3.2.1.1 Sets of parts of speech and features

The final set of categories and features in the main lexicon is different to the traditional or any other classifications. The basic *pos* set was the one in SMORPH, modifying it *to facilitate the creation, in the simplest way, of a (simplified) model of the Spanish language, as a sequence of categories described with a context free grammar*.

According to this criterion, if a word or set of words has the same lexical behavior than other set of words in the same category, a new category is not justified and thus not used (even if it is a traditional category). Words belonging to the same traditional category, but with a different behavior will be separated in different *pos*.

For example: all the different types of determinants have been joined in a single category because it has been verified that they take part in the same structures (the differences are modeled with features).

However, other set of words, normally grouped as prepositions, have been separated in three groups: *al* and *del* in a group, because they are prepositions plus determinant, *a* and *de* because they cannot appear in some structures (they are transformed in *al* or *del*), and the rest of the traditional prepositions.

The study has led to complete the classification in categories, shifting part of the expressive load from the categories to the features, with our particular method. The chosen features make it possible to indicate the gender, number, etc., of each word. The advantages of this management will be shown in the section 3.3.3 (page 25) about formal grammars. Not all the features are the same than the traditional ones because they have been chosen to model specific behaviors in particular structures.

The list of categories and the features that can be applied to each one are described below:

- Verbo (verb). The same than the one used in the traditional classification. It admits the following features:
  - Tipo de verbo (verb type): regular, *copulativo* (a specific category for the verbs ser (to be), estar (to be), parecer (to look like) and some others), verbs that are followed by: past participle, gerund, infinitive, *que* plus verb in flexive form, *que* plus verb in *subjunctive, de* plus infinitive, *a* plus infinitive.
  - Tipo de terminación verbal (sort of verb ending): infinitive, past participle, gerund, flexive form. The flexive verbs have other feature, the *mode*: indicative, subjunctive, conditional, imperative.
  - Tiempo verbal (verbal tense): present, imperfect, preterit, future.
  - Persona (person): 1, 2, 3.
  - Número (number): singular, plural.
  - Género (gender): feminine, masculine
- Nombre (noun). The same than the one used in the traditional classification. It admits the following features:

- Número (number): singular, plural and neuter.
- Género (gender): feminine, masculine and neuter plus an extra value for the nouns starting with a stressed *a*.
- Tipo de nombre (type of noun): común (common), propio (proper name).
- Nombre seguido por subjuntivo (name followed by a subjunctive). It marks names that admit the structure *que* plus a verb in subjunctive.

- Adjetivo (adjective). The same than the one used in the traditional classification. It admits the following features:
  - Género. (gender): masculine, feminine and neuter
  - Número (number): singular, plural and neuter.
  - Tipo de adjetivo (sort of adjective): comparatives or not.
  - Adjetivo superlativo (Superlative adjective).
  - Tipo de adjetivo según la posición (sort of adjective, depending on the position it may take with respect to the noun): only after it, only before it and adjectives that may occupy any position.
  - Adjetivo que impone un modo verbal (adjective that impose a verbal mode. For example, subjunctive or indicative).
- Determinante (article). It includes the articles and some adjectives (possessive, demonstrative, etc.). The differences in the behavior in some structures (specially the ones involving sequences of *articles)* are modeled using features, which allows restricting some of them to some particular positions. The set of features that it admit is:
  - Género (gender): masculine, feminine and neuter.
  - Número (number): singular, plural and neuter.
  - Determinante que admite estructuras comparatives (article that admits comparative structures).
  - Tipo de determinante (type of article defined of indefined).
    - Type of defined articles (articles, demonstrative, relative, possessive).
    - Type of indefined articles (indetermined, cardinal, interrogative).
- Pronombre (pronoun). Groups the words that may be the head of a nominal phrase. They mainly coincide with the traditional classification. The features that may be applied to them are:
  - Género (gender): masculine, feminine and neuter
  - Número (number): singular, plural and neuter.
  - Tipo de pronombre (type of pronoun): personal or impersonal.
    - Some personal pronouns are marked as “cannot follow a preposition”.

    - Impersonal pronouns are divided into defined and undefined.
        - Defined impersonal pronouns are divided into determined and demonstrative.
        - Undefined pronouns are divided into undetermined, cardinal and interrogative.

- Pronombre que sigue a una preposición (pronoun that follows a preposition). It only includes 3 words, that are pronouns that may follow a preposition: *mí, ti* y *sí*. They cannot follow the preposition *con.*
- Pronombre más preposición (pronoun plus preposition). This category only includes *conmigo, contigo* and *consigo,* because of their particular behavior, as explained in the previous paragraph.
- Clítico. It includes a subset of the pronouns that can only be located before the verb. They can appear isolated or in sequences of two *clíticos.* In this last case, the order may be considered (not all of them can follow any other). This has been modeled with a new feature that indicates if the clítico may only occupy the first or the second position, or any.
- Ordinal number.
- Preposición (preposition): This category does not contain exactly the set of words that are traditionally included in the preposition category. Some words, with a similar behavior, have been added, and others have been excluded, because of their particular behavior in some structures. No features have been added to this part of speech.
- Artículo contracto (contracted article). It includes *al* and *del (*preposition plus article masculine singular).
- Prepositions *a* and *de.* This category only contains the words *a* and *de,* because they cannot be followed by an article masculine singular. In this case they are transformed to the previous category.
- Adverbio (adverb). The same than the one used in the traditional classification. The are marked with a feature to indicate that they can be used in comparative structures or not.
- Conjunción (conjunction). The same than the one used in the traditional classification. They are marked with a feature that indicates whether they are joining phrases of the same or different syntactic level.
- Interjección (interjection). The same than the one used in the traditional classification although in the dictionary we have only included a small number of them (because of their low frequency) composed only by a single word.
- Cajón de sastre (miscelaneous). It contains a set of words difficult to classify, whose behavior does not coincide with any other word and is not homogeneous. It contains words that take part in some concrete expressions, and do not appear out of them. In the formal grammar every word in this category has its particular rules to model the expressions in which it appears.

- Signos de puntuación (punctuation marks). It contains all the punctuation marks except the full stop.
- Full stop. It only contains the full stop, as the element to separate sentences.

3.2.1.2 Main lexicon management mechanisms

The general lexicon is used in two different processes: the prediction, and the tagging of written words. Because of its big size, any algorithm that implies scanning all the words in the dictionary would be very slow. In order to solve this problem, auxiliary mechanisms have been included in the management of the dictionary. These mechanisms imply the use of:

- Hash indexes, for direct access to each word in the categorization process.
- Intermediate indexes arrays that reduce the number of words to scan to the ones that satisfy certain constrictions such as starting with a particular letter or pair of letters and/or belonging to a particular category.

3.2.1.3 Main lexicon limitations

The main lexicon has two main problems:

- Its vocabulary and frequency information are adapted to the training texts (journal texts).
- It is static, so that it is not adapted to the user or the subject.

These characteristics make the dictionary very adequate for writing newspaper texts, but the prediction is not optimal when writing texts about any other subject or with a different style. For this reason, the adaptive lexicons that will be explained in the following sections have been included in the system.

3.2.2 Personal and subject lexicon

The *personal lexicon* is generated while the user writes, storing all the vocabulary, the frequency of each word in that text and the bigrams of each word. This lexicon is generated in every session, so that at the beginning it is empty, and at the end of the session it achieves the best performance, because of the adaptation of the vocabulary and frequencies.

A *subject lexicon* is a dictionary generated from any text or set of texts about a particular subject. In the subject lexicon the same information than in the personal dictionary is included. Several subject lexicons can be generated, but only one of them can be active in a certain moment.

The personal and subject lexicons are trained using plain text (not necessarily tagged).

3.2.2.1 Problems of the personal and subject lexicons

Although the learning capability makes the performance of the personal lexicon better than the one of the main lexicon, it may be a problem for some users.

The adaptability of the lexicon makes it learn all the words the user writes. This is especially good, for example, when the user writes technical documents that include specific vocabulary that may not be included in the main lexicon. The problem arises with users who frequently misspell words, because they are learning to write or have linguistic problems such as dyslexia or because they are not very accurate with the keyboard or with the switch. These wrong words would be included in the dictionary and predicted, shifting correct words in the menu and reinforcing the mistakes. For this reason, two control mechanisms have been included in the architecture:

- Removing of the learning capability for new words. The personal dictionary will adapt the frequencies and bigrams *in case the word is included in the main lexicon.* Otherwise, the word will be ignored. The performance of the whole system will improve, for the adaptation of the frequencies and the bigrams, but new words will have to be typed each time the user desires to write them. This control mechanism is specially useful for persons with very frequent mistakes.
- People with sporadic mistakes may include a less drastic method that learns new words but *not immediately:* the new words are included in the dictionary, but they will not be predicted until the user writes them at least a minimum amount of times. If mistakes are sporadic, misspelt words will not be written several times, and will never be predicted.

3.3 Prediction methods

As it has been explained in the previous section, the main, personal and subject lexicons include several types of information: probabilistic, grammatical and some on the user style. Next, the different prediction methods that utilize all this information are explained. For each method, we will explain its advantages and disadvantages, the information sources it needs and the working procedure.

The first one is the method based only on probabilistic information and more complex information sources will be added, such as probabilistic grammatical information and a context free grammar.

3.3.1 Basic probabilistic methods

The simplest prediction method is based on the use of *unigrams* (the absolute frequency of the word). It only needs the beginning of the current word and predicts the more probable words starting with that letters according to the corresponding lexicon. When this method applied to the main lexicon, the predictions obtained after a blank space (or a particular group of letters) are always the same, independently of the user, the subject, or the position in the sentence, because that lexicon is static. When applied to the personal

lexicon, the predictions change through the text, because the lexicon is trained simultaneously to the writing process.

This method obtains up to three different lists (one from each lexicon), that will be combined by the management module, as will be explained in the section 3.4 (page 38).

The amount of information taken into account may be augmented, making statistics of words sequences, obtaining *bigram* (for sequences of 2 words) and *trigram* models (for 3 words). Then, the prediction method considers, not only the beginning of the current word, but also the last word or pair of words.

The main problem of these methods is the amount of text needed to train them, that has to be big enough to ensure that each valid word sequence appears a relevant amount of times. They also need great amounts of computational resources in the training and the working stages, especially if the number of words in the lexicon is big or the desired sequences are bigger than 2 or 3 words.

For these reasons, sequences of only two words have been considered in our word prediction system, exclusively in the personal and subject lexicons. In the main lexicon, bigrams or trigrams are not considered because of the huge amount of resources needed to manage the valid sequences of the 130000 different words it contains.

3.3.1.1 Limitations of the basic probabilistic methods

Low performance of the prediction based on *unigrams* is mainly due to the small amount of information it considers: it does not take into account the relationships of each word with the previous ones, so, sometimes the predictions do not make sense in the sentence context. *Bigrams* and *trigrams* consider more context information but they require a huge amount of training information or the performance will decrease. In our prediction system, this problem has been solved grouping the words in categories (parts of speech), according to their behavior, given rise to the models explained in the following section.

3.3.2 Probabilistic *pos* models

These models are based on the statistics of the sequences of parts of speech. They solve part of the problems of the basic probabilistic models, because they require less training text. They also include short-term grammatical context information in the prediction process because they prioritize the more probable words *belonging to the pos that more probably follow the pos of the previous word or words*. The main problem of these methods is the information they need: the grammatical information of each word must be included in the system, as well as the relationship among the different *pos* (the probability of each sequence).

The set of *pos* used is described in the section 3.2.1.1. It may not be the optimum for a statistical *pos* model, but, in this case, the criterion to optimize the formal model has been prioritized.

In our system, *bipos* and *tripos* models are included, and have been trained from a 55.000 word text tagged without ambiguity. As there are less than 20 categories, this text is

enough to train the *bipos* model, but the *tripos* matrix has a high amount of *zeros* (sequences of *pos* that do not appear in the training text) that influence the results, as will be shown in the chapter or results.

In the prediction process, the list of words that will be shown to the users are the ones with higher probability according to the following formula:

$$p(w_t / w_{t-1}) = \sum_{\forall C_{t-1}^j \in C_{t-1}} \sum_{\forall C_t^i \in C_t} p(w_t / C_t^i) \cdot p(C_t^i / C_{t-1}^j) \cdot p(C_{t-1}^j / w_{t-1})$$

Equation 1. Probability of the word w_t in the bipos model.

Where:

- $p(C^i_t/C^j_{t-1})$ is the probability that category C^i_t follows the category C^j_{t-1}. It is obtained from the *bipos* matrix.
- $p(w_t/C^i_t)$ is the probability to predict w_t given the category C^i_t. It depends on the absolute frequency of the word in the considered dictionary (among other factors), so, it will be different when the method is applied to the different dictionaries. This method will be applied to all the lexicons, and the management module will select the final predicted words.
- $p(C^j_{t-1}/ w_{t-1})$ is the probability that the word w_{t-1} belongs to the category C^j_{t-1}. This information is included in the main lexicon.
- C_{t-1} and C_t are the sets of categories of the words w_{t-1} and w_t respectively (both can belong to several categories, and it must be considered in the prediction).

Equation 1 will be applied to all the words in the lexicons, and the more probable ones will be selected, and presented to the user as predicted words. The lists of words predicted after a blank space or any group of letters will be different, depending on the *pos* of the previous words written in the text, because a high word frequency may be reduced by a small *pos* probability and vice versa. The result is that the most frequent words belonging to the most probable *pos* are predicted.

This method can be extended to include in the prediction as many previous words as we desire. Because of limitations in the amount of tagged text to train it, the current system considers a maximum of two previous words in the prediction (*tripos* model).

3.3.2.1 Improvements in the probabilistic *pos* models

3.3.2.1.1 Features management

The prediction methods based on *bipos* and *tripos* predict the grammatical categories of the next word. In our system, a great amount of information about the words is contained in the features. This reduces the number of categories, somehow degrading the performance of the methods based on *pos.*

A post-process has been included in the prediction process to filter some words depending on the values of their features and the features of the previous words, so the list of words is more adequate in some cases. As there are a great amount of features, not all of

them have been included in this process, but only the most frequently used ones (in the sense that the highest number of words have them), which are gender and number. The process is applied to nouns and adjectives, when they follow an article, *artículo contracto,* adjective or noun, and checks whether the gender and number are compatible, rejecting words that do not satisfy this condition.

This is an approximate method that sometimes filters words that should not be eliminated, but its global effect is positive, and it improves the performance of the prediction, as will be shown in the chapter on evaluation results.

3.3.2.1.2 Techniques of matrices smoothing

One of the main problems in these models is the lack of training data to calculate accurately the probability matrices. One of the methods used to reduce its negative effects is the smoothing of the *bipos* and *tripos* matrices, in order to eliminate the *zeros* included in them. Several smoothing methods have been tested, obtaining better results than methods without smoothing.

However, the method with better results (as will be shown in the chapter of results), consists in using the tripos matrix without smoothing, and, in case a *zero* is found in the desired position, use the bipos matrix (fall back strategy). If the list of predictions is not long enough, it is completed with predictions from the method based on unigrams.

3.3.2.1.3 Training with text tagged with ambiguity

Some tests have been made, training the bipos and tripos matrices from text tagged automatically. The advantage of this method is that this text is very easy to obtain, especially when comparing with the effort needed to get manually labeled text. In the automatically tagged text, each word is associated with all its possible categories, and the probability of each one. When training the matrices, each sequence of words increment the counts of several matrix positions, depending on the probability of each word to belong to each category.

The results obtained using these matrices are not as good as the ones obtained with matrices trained without ambiguity, but are not far from them. In some cases (when the test and training texts have the same style) some results of the *ambiguous matrices* have been better. This may be due to the reduction in the number of zeros produced by the ambiguity and the increase in the amount of training texts used, although, if it is available, unambiguously tagged text should be used.

3.3.2.1.4 Unknown words management

One of the problems of the pos methods is the presence of *unknown words* in the texts. These words are not included in the dictionary, so that the system does not know their grammatical information, and can not apply any grammatical prediction method.

There are a series of reasons that ensure that new words will always appear in the texts: the amount of words in a language (much bigger than the size of any lexicon), new

words (for example, technical terms), use of words of other language, family names, slang, mistakes, etc.

In order to handle all these cases, a new module has been included to assign a *pos* to the new words the user writes. This automatic tagger is based on the use of suffixes, prefixes, and an automatic module that conjugates regular verbs. It is considered whether the prefix or suffix change the category of the original word or keep it, and whether they have information of the features or not. The regular verb *conjugator* recognizes any form of a regular verb[*], finding out its lemma, category (verb) and features, even if it has attached *clíticos* or the verb form has *tilde*. It also assigns *pos* to numbers and punctuation marks not included in the dictionary.

All the categorization processes are combined recursively, so, for example, it is possible to recognize a regular form of a verb with two *clíticos* and a prefix.

This automatic tagger assigns pos and features to words that are not in the main lexicon, but keep a relationship with words included in it. However, there will be words that still keep untagged. In order to assign a *pos* to these words, statistics have been made, taking the 55000 word manually tagged text from newspaper and tagging it again with this system. Studying the words that the system could not categorize and their *pos in the first text* the results were:

Noun	74%
Adjective	23%
Adverb	1.5%
Verb	1.5%

This is the set of *pos* that are assigned to unknown words in the system. The features assigned are the more flexible ones (less restrictive): neuter as gender and number of nouns and adjectives.

Obviously, none of this *pos* is any of the *closed pos*[†]. The distribution corresponds with the style of the journal: a high amount of nouns (people and places) and some new adjectives. The small amount of verbs is due to the complete verb representation we have included in our system.

Obviously, this system assigns a tag to any word, even mistakes or words of other languages, but the global effect when it is included in the prediction is good, as it is shown in the chapter on evaluation results.

3.3.2.2 Limitations of the probabilistic *pos* models

These methods are based on short fixed sequences of categories, usually 2 or 3. This may be a very small number in some structures, because the category or features of a word may depend on other word or words separated by more than 3 positions. These long

[*] Forms of regular verbs are not included in the main lexicon to reduce its size.

[†] Closed category: *pos* that includes a small number of words. All the words belonging to closed categories are included in the dictionary.

term relationships cannot be modeled with the bipos or tripos, because of the limited context information they consider.

This is the reason why more sophisticated models may be included in the system: the formal models.

3.3.3 Formal grammars

Formal models are used in the prediction in order to include long term relationships among the words. After a detailed analysis of the models presented in the section 2.1, we have selected a *context free grammar* because of its working method (which is adequate to model our problem), expressive power, computational resources needed, available information, parser complexity, etc.

The set of *pos* and features to use in the rules are the ones explained in section 3.2.1.1 (page 16). Each category and feature has been selected to optimize the number and expressive power of the rules, so that they allow to build a simplified model of the Spanish language, as a *pos* sequence described with a context free grammar, in the easiest way. Each word in the main lexicon has been tagged with the adequate category or set of categories, and features, so that this information is available when the grammatical modules demands it.

3.3.3.1 Parser selection

Among the different parsers that are able to analyze a context free grammar, we have selected an Earley parser, taking into account the following considerations:

- Our problem constraints: the sentence must be parser from left to right, because only the left part is available.
- Its predictive power.
- It is possible to keep several branches in parallel, with different valid parsing possibilities, allowing the management of ambiguous sentences.
- It is not necessary to restart the analysis when a branch fails.
- It is possible to add frequencies and features management, increasing the parser power.

The Earley parser is powerful and flexible enough to model a (simplified) natural language. In [Stol95] and [Earl70] a description of the basic algorithm can be found.

3.3.3.2 Modified parser

Now, an example of the analysis process is presented, adding the modifications needed to build a prediction system from the parser. The example sentence is *"La casa está en la colina" (the house is on the hill).*

Notation: *non terminal symbols* (those generated from the rules) will be written as a sequence of capital and small letters, and *terminal symbols* (the set of *pos)* as a sequence of small letters.

The set of rules is:

(R1) S → NP verb Sprep (Sentence. Initial symbol)

(R2) NP → noun

(R3) NP → art noun

(R4) NP → art adj noun

(R5) Sprep → prep NP

Firstly, the parser predicts (because of its top down parsing strategy). The initial symbol is expanded, applying R1. As it expects a NP, all the rules whose left hand part is NP are applied. In each rule, a circle (°) is included, marking the position where the analysis is stopped.

(R1) S → ° NP verb Sprep

(R2) NP → ° noun

(R3) NP → ° art noun

(R4) NP → ° art adj noun

R1 stops the analysis until a NP is completed. R2, R3 and R4 predict nouns and articles, so the prediction list will be composed by nouns and articles.

The word "*la"* is included in this list and is selected by the user. "La" can be an article, pronoun or a noun, so it extends the three rules:

(R2) NP → noun °

(R3) NP → art ° noun

(R4) NP → art ° adj noun

As it can be observed, the ° is in the final position of R2, indicating that the analysis has finished, and a complete NP has been formed (although the other possible NP branches are still active). Then R1 is expanded, keeping a copy for R3 and R4. After these step, the set of active rules are:

(R3) NP → art ° noun

(R4) NP → art ° adj noun

(R1.1) S → NP ° verb Sprep

(R1.2) S → ° NP verb Sprep

This set of rules predicts nouns, verbs and adjectives. The following word is "*casa"* that can be a noun and a verb. When it is included in the analysis, R4 is eliminated, the verb extends R1.1, that, then, expects a Sprep. The noun expands R3, finishing it, and the obtained NP extends R1.2. After this process, the active rules are:

(R1.1) S → NP verb ° Sprep

(R1.2) S → NP ° verb Sprep

(R5) Sprep → ° prep NP

This set of rules predicts verbs and prepositions. The analysis continues in a similar way, until the input sentence finishes. If, at the end of the sentence, an analysis exists with the ° in the final position of any of the rules that expands the initial symbol (R1 in this example), the sentence is grammatically correct. The sentence of this example is correct, because at the end, the analysis of one of the interpretations is:

(R1) S → NP verb Sprep °

As we have seen, the parser analyses from left to right, keeping all the possibilities, and it is not necessary restart the analysis when a branch fails.

The base for the development of the predictor has been the system of tables used by *Bison,* a parser generator. *Bison* takes a grammar in the adequate format as input and generates a parser for that grammar. *Bison* source code has been modified to change the information it process and generates, and new elements have been added. After that, a new system has been developed to use that information to predict instead of analyze. The power of this new system will be explained in the following sections.

3.3.3.3 Improvements to the basic parser

Now, the main contributions in the formal grammar are described, mainly, the description of the improvements in the expressive power of the parser and the rules. These contributions make the parser more adequate to describe the linguistic phenomenon we have considered relevant after the study of text and linguistics. For each new feature added to the parser, it is provided: its description, the justification of the necessity, the performance they produce (as a reduction in the number of rules and categories needed to describe the same linguistic phenomenon) and the procedure to include it in the prediction process.

In the example, the definitive format of the rules is used. The basic format is the one used in *Bison,* powerful enough to model a conventional context free grammar. *Non terminal symbols* (those generated from the rules) will be written as a sequence of capital and small letters, and *terminal symbols* (the set of *pos)* as a sequence of small letters. A basic rules is:

NP:

det adj n adj

;

In case several rules expand the same non terminal symbol, they are separated by a "|":

NP:

det n

| n adj

| pron

;

In each section, the format (syntax) to include the information necessary for the new features included in the parser are described.

3.3.3.3.1 Management of the probabilistic information

The probabilistic information allows to ponder the information of the different branches that correspond to the various possible interpretations of the current sentence. The branches can be due to the different meanings of the words and the different structures that can correspond with the written part of the current sentence. The probabilistic information allows to give more importance to the predictions made by the more frequent rules, and to the branches followed by the more common meanings of the words.

Example:

NP:

det n & 750

| n adj & 100

| pron & 150

;

Where the symbol & precedes the probability of the rules (multiplied by 1000). In this example, the nominal phrase is composed by an article plus a noun (75%), or a noun followed by an adjective (10%) or a pronoun (15%).

This probability can be trained, counting the amount of times each *non terminal symbol* is used, and each rule that generates it. In order to do this, it would be necessary to have a text tagged with the parts of speech and the rules. As this text is not available, an approximate method has been used, that allows to calculate the probability from an ambiguously tagged text. It counts all the rules that could be used, pondering them with the probability of the words to belong to the categories needed to follow each particular rule. This is an approximate method, but manual calculus have been made, on a subset of the same corpus, obtaining similar values. The number of parameters to calculate in the current version of the rules is approximately 260.

In the future, other training methods such as the Input-Output algorithm explained in [Pere91] will be included, modifying them to include our specific features specific.

The probability of the next category depends on the probability of the branches that predict that particular category, the frequency of the rules applied in each branch, and the probability of each word to belong to the category needed to follow the particular branch. The formulas to apply are:

$$p(Branch) = \prod_{BRr} p(Rr) \prod p(C^{k}_{t-j} / w_j)$$

$$p(C_t^i) = \sum_{B_{C_t^i}} p(Branch)$$

$$p(w) = \sum_{C_t^i} p(w / C_t^i) \cdot p(C_t^i)$$

Equation 2. Set of equations to obtain the probability of the word w in the parser based method.

Where $B_{C_t^i}$ are the branches that expect the category C_t^i and BR_r are the sets of all the rules R_r extended in the branch r. C_{t-j}^i is the category *i* of the word w_j^i.

This formula is applied to all the words in the lexicons and the more probable ones are selected to be included in the prediction list. These formulas allow to prioritize the words predicted by the most probable rules, followed by the most probable categories of the words.

3.3.3.3.2 Management of optional symbols

Some structures in natural language are valid with and without a particular symbol. These structures can be handled using different rules for each one, but the power of the parser has been increased to manage them with a single rule. **Any (terminal or non terminal) symbol, may be optional, and as many optional symbols as necessary can be included in any rule (but not all of them).**

In the rule, optional symbols are marked with a *?* plus the probability to appear in the sentence.

```
NP:
        det adj ? 400 n                 & 1000
;
```

This rule models nominal phrases with or without adjective before the noun. The adjective appears 40% of times. The parser expands a single rule at the beginning, processing it in a normal way until the optional symbol. Then it duplicates the rule, advancing an extra position in one of its copies. Then, one of the rules predicts the optional symbol, and the other the following one, with the adequate probabilities management as can be observed in the following example:

```
NP:
        ° det n adj ? 400 Sprep ? 300           & 1000
        ;
```

After writing a *det* and a *noun* the rule is duplicated:

```
NP:
        det n ° adj Sprep ? 300         & 400
        det n adj ° Sprep               & 180
        det n adj Sprep°                & 420
```

;

The first rule predicts an adjective, with the probability that the adjective appears in the sentence. The second one will expand the rules of the Sprep, with the probability that the adjective does not appear, times the probability that the Sprep does. The last one finishes the NP, with the probability that none of the optional symbols is written.

In the current version of the parser the probability of the optional symbols is trained with a procedure similar to the one used to train the probability of the rules from ambiguously tagged text.

3.3.3.3.3 Management of words and lemmas

Some structures in natural language depend on a particular word or a lemma (family of words), for example, *ir a + infinitivo (go to + infinitive),* that needs any form of the verb *to go* plus the word *to* plus a verb in infinitive. To model that behaviour using a context free grammar, a *pos* containing only that word or lemma will be needed, to be able to write the rule for that concrete structure. It can be done in two different ways:

- Adding a category for that word, taking it out of the *pos* where it was included. Then, it is necessary to duplicate all the rules of the original *pos* for the new one, so that the word can still appear in the structures of its original *pos*.
- Add a category for the word, keeping it also in its original *pos.* In this case, there is a problem with the probability of the word to belong to each *pos:* what is the probability of a word to be itself and to be a noun, for example?

Other words normally follow the rules of its *pos*, except in some particular cases where it be handled in similar ways, with the same kind of problems.

For these reasons, the power of the parser has been increased again, so that it is possible to model not only sequences of categories, but also exceptions where particular words or lemmas have a different behavior. **The new parser allows imposing or prohibiting words or lemmas** at any point in the rule, without segregating the words to a different category**.** This new feature reduces the number of categories needed in the grammar, because no new *pos* is necessary to model the structures and also reduces the number of rules, because they do not have to be duplicated for the new *pos,* thus eliminating the probabilities coherence problem, because the segregation is not necessary.

Example:

Expr1:

prep [sgte{for}]

noun [sgte{example}]

;

To model the expression *por ejemplo (for example)* separating the words in different categories, we would have to write the rule for *for example,* and then duplicate all the rules of the prepositions for the new category of *for* and all the rules of the nouns for the new *pos* of *example.*

The imposition of lemmas or words imply a different behavior of the rules. In case a word is imposed, the rule in that point predicts only that word, and accepts only that particular word as being valid. When a lemma is imposed, the prediction of the rule in that point will be words with that lemma, and only them will be considered valid ones.

The probability management changes: when a word is imposed, the probability of the rule is applied to the predicted word:

$$p(w) = \sum_{C_t^i} p(w / C_t^i) \cdot \sum_{B_{C_t^i}} p(Branch_{C_t^i}) + \sum_{B_w} p(Branch_w)$$

Equation 3. Probability of the word, adding the component to handle the imposition of words.

Where B_w and $Branch_w$ are the branches that predict the word *w.*

Branches that impose a lemma, include a new term in the formula, for its particular behavior: the probability of the rule is divided among all the words with that lemma, so the probability of a word fixing a particular lemma is:

$$p(w / Lemma_w) = \frac{\#(w \cap Lemma_w)}{\# Lemma_w}$$

where $\#Lemma_w$ is the sum of the frequency of the words whose lemma is *Lemma*, and $\#(w \cap Lemma_w)$ is the sum of the frequencies of the words whose lemma is *Lemma*, and whose exact word is *w* (several different entries may coincide in both fields).

$$p(w) = \sum_{C_t^i} p(w / C_t^i) \cdot \sum_{B_{C_t^i}} p(Branch_{C_t^i}) + \sum_{B_w} p(Branch_w) + p(w / Lemma_w) \cdot \sum_{B_{L_w}} p(Branch_{L_w})$$

Where B_{L_w} are the branches imposing $Lemma_w$.

The prohibition of lemmas and words is managed in a different way. The prohibition is indicated with the symbol ~ preceding the word or lemma. For example:

Sentence1:

```
Sentence
prep    [sgte{~bajo ~hacia}]
conj    [sgte{que}]
Sentence &1000
;
```

This rule models complex sentences, joint with a preposition plus *que (what)*, where not all the prepositions can occupy the second position, so *bajo (below)* and *hacia (towards)* are prohibited. In case we desire to prohibit a lemma, the format will be similar, indicating *lema* instead of *sgte*.

The prohibition management is handled in a different way. It is included as a filter in the prediction process: when the lists of words proposed by the rule is generated, the prohibited words (or the words whose lemma is prohibited) are eliminated from the list. In

case the prohibited word is written by the user, it will not be accepted by the rule, making it fail, although this word may be considered valid by other active rule.

3.3.3.3.4 Management of features

As it has been explained in previous sections, a great part of the grammatical information associated to each word is included in the features, not in the *pos.* The main advantage of this point of view is the reduction in the number of categories and rules in the language model, when compared with other grammar with the same power, based only on categories.

If we desire to model a nominal phrase with three components, an article, a noun and an adjective, that agree on gender (masculine, feminine and neuter) and number (singular, plural and neuter), using conventional *pos* in the grammar, different *pos* will be necessary for nouns masculine singular, and nouns feminine singular, etc., so that the different valid combination of features have to be included as different *pos.* This implies an increase in the number of categories from 3 (article, noun, adjective) to 27 (noun $_{\text{singular-masculine}}$, noun $_{\text{singular-feminine}}$, etc.). The number of rules needed to model some of the structures in a noun phrase increases from 1 to 200, to ensure the agreement in gender and number.

The features of each word influence the features of the following words in several different ways: imposing or prohibiting a particular value, or demanding agreement in any feature. The parser has been modified to support all these features.

3.3.3.3.4.1 Features agreement

The agreement consists of ensuring that the values of some particular features of different words coincide. For example, in Spanish, all the symbols in the nominal sentence (articles, adjectives and nouns) have to agree on gender and number, and in number with the verb of the sentence.

The parser has been improved to support the agreement. It is flexible enough to allow selecting the feature or set of features that may agree, and of which words. Any feature may agree in any word, and a particular word may agree in a feature with a word or words, and in other feature with other word or words. The agreement is *intelligent,* in the sense that it combines all the information in the features of the implied words: for example, in case three words may agree in gender, and the first one is neuter, when predicting the second one, no restrictions are made in the gender, and, in case the second is masculine or feminine, the adequate restrictions will be made when predicting/accepting the third word.

An example of nominal phrase where there are several features that should agree, using the format of the system, is:

NP:

det	[gen{X} num{Y}] ? 175
adj	[gen{X} num{Y}] ? 500
n	[gen{X} num{Y}]

adj [gen{X} num{Y}] ? 250

prep

det [gen{Z} num{M}] ? 75

n [gen{Z} num{M}]

;

This rule considers the following agreements:

- Gender and number of the first four symbols, using X and Y.
- Gender and number of the last two symbols, using Z and M.
- The prepositions do not have to agree with any other symbol.

As it can be observed, the articles and adjectives are optional, but, in case they appear they have to respect the agreement constraints.

The agreement constraints are included as filters in the prediction process, checking if the word agrees with the previous ones (in case there is any), and eliminating the ones that do not agree. If the user writes a word that does not agree, it makes that rule fail (although the word may be valid for other active rules).

The separation of the grammatical information in categories and features and the improvement in the parser capabilities reduce drastically the number of *pos* and rules to model some structures. In this system, a nominal phrase with three symbols agreeing in gender and number can be modeled with a single rule, whereas the conventional system requires up to 27 categories and more than 200 rules.

3.3.3.3.4.2 Imposition of a concrete value of a feature

Some words do not require agreement with the features of other word, but need that the features of the other word have a concrete value. For example, the verb *to have* must be followed by a verb in *past participle.*

The parser has been modified to allow imposing a value to any feature of the symbols of the rule. For example (the equivalent in English):

VerbalForm:

verb & 700

| verb [lemma{go}] prep [sgte{to}] verb [verb_type{inf}] &100

| verb [lemma{have}] verb [verb_type{past-partic} gen{masc} num{sing}] & 200

;

This rule models the verb formed by a single word, *have + past participle,* and the periphrasis *go to + infinitive.* In it, it can be observed that the verb *to have* needs a past participle, masculine, plural, and the verb *to go* requires an infinitive form.

To impose a value to a feature, to one or more symbols in a rule, with the current system, a single rule that includes this constraint is needed. The conventional system would

require to repeat the rule for each valid combination of the other features: for example, to impose that a particular noun is singular, six rules will be needed: noun $_{\text{singular-feminine}}$, noun $_{\text{singular-masculine}}$, noun $_{\text{singular-neuter}}$, noun $_{\text{neuter-feminine}}$, noun $_{\text{neuter -masculine}}$, noun $_{\text{neuter -neuter}}$. In case the singular nominal phrase consists of two words, 16 rules will be needed, and 32 rules in case it is composed of 3 elements.

The imposition of features is included in the prediction process as a filter that eliminates the words that do not contain the imposed feature. In case the user writes a word without the feature, the rule will be stopped.

3.3.3.3.4.3 Prohibition of concrete values of a feature

Some language structures require that a word has any feature value *except for* a particular value. For example, in Spanish, adjectives (in general) can appear before and after the noun, but some of them can only appear before or after it. To model structures like those, the adjectives that cannot appear in one of the positions are marked with a feature that is prohibited in that point of the rule.

NP:

```
   det    adj [adj_type{~a3}] ? 100    n    adj [adj_type{~a1}]  ? 100    & 1000
;
```

This rule models a nominal phrase with two adjectives, and each one (in case it appears) may be located in the right position: the first one cannot be an *a3* adjective (*a3* adjectives can only appear after the noun) and the last one cannot be an *a1* adjective (*a1* adjectives can only appear before the noun).

Of course, this could also be solved separating the adjectives in different categories, thus increasing the number of categories and rules needed to model this structure. In case it is desired to prohibit a particular value of a feature in a structure, using conventional models, it would be necessary to list all the non-prohibited combinations of features.

The prohibition of features is included as a filter that eliminates the words that contain the prohibited feature from the prediction list.

3.3.3.3.4.4 Management of features in non-terminal symbols

In previous sections, the management of features in terminal symbols has been shown. It is also important to *transmit* feature values from the non-terminal symbols, and to them, for example, when several non-terminal symbols in different points of a sentence may agree. In this case it is necessary to be able to find out the features of the non-terminal symbol from the features of the terminal symbols included in it, or using other methods. It is also necessary to define the way the values of the features of the non-terminal symbol affect the features of the symbols that compose it.

The parser has been modified to support the non-terminal symbol features agreement. The features of each non-terminal symbol will be defined in the rule, and used afterwards in the same way as if they had been included in the lexicon.

For example: the rule to model *la casa es bonita* would be

S:

NP [gen{X} num{Y}]

v [lemma{ser} num{Y}]

adj [gen{X} num{Y}]

;

where:

NP [gen{X} num{Y}]:

det [gen{X} num{Y}]

n [gen{X} num{Y}]

;

At the beginning of the sentence, a NP is needed, so *det* are predicted. As it is the first word, X and Y are empty, and no restrictions are imposed on them. When the user writes the article, X and Y have its values, and the features or the next word may agree with them. When the noun is written, the NP will be complete, with the values of the features of the words included in it (the more restrictive, in case one of the words have neuter gender or number). After that, the verb is predicted, requiring that the number agrees with the NP.

If the NP were the following one, the gender and number of the NP would only depend on the features of the first words:

NP [gen{X} num{Y}]:

det [gen{X} num{Y}] ? 175

adj [gen{X} num{Y}] ? 500

n [gen{X} num{Y}]

adj [gen{X} num{Y}] ? 250

prep

det [gen{Z} num{M}] ? 75

n [gen{Z} num{M}]

;

There are also special cases, where the values of the features do not depend on the values of the features of the non-terminal symbol and therefore are not extracted from the values of the components. For example

NP [gen{feminine} num{sing}]:

det [gen{masculine} num{sing}] ? 175

n [gen{a-tónica} num{ sing }]

adj [gen{feminine} num{ sing }]

;

This NP uses a particular type of noun that starts with *a* and requires a masculine article, but agrees with feminine adjectives within the NP and in the rest of the sentence, so, the features of the NP are feminine singular. A nominal phrase with these nouns will be *El agua limpia (the clean water).*

It is also possible to impose or prohibit a value of a feature and its agreement with another one, so that the rule allows controlling the agreement with *external* symbols, *in case it is compatible with the prohibited or imposed value.* In case they are not compatible, the analysis of that rule will stop. For example:

```
NP      [gen{X} num{Y} st-prondet{Z} person {3}]
        pron [gen{X} num{Y} st-prondet {Z} st-prondet{~ determined }] & 1000
        ;
```

This nominal phrase extracts the gender and number from the value of the features of the pronoun. It will be always third person (it is not necessary to extract it from the pronoun), and it *cannot* be a determined pronoun, but it has to agree with the value in Z, so, if the value of Z is *determined,* this rule will fail.

The possibility to handle features in non-terminal symbols makes it possible to control them. This capability is complemented with the possibility of using rules in a recursive way, for example:

```
VerbalForm [verb_type{X}]:
        verb [verb_type{X}] & 700
        | verb [lemma{go} verb_type{X}] prep [sgte{to}]
          VerbalForm [verb_type{inf}] &100
        | verb [lemma{have} verb_type{X}]
          VerbalForm [verb_type{past-partic} gen{masc} num{sing}] & 200
        ;
```

This rule allows us to model any periphrasis formed by a *have + any verbal form starting with a participle* and *go to + any verbal form starting with an infinitive*. Verb_type{X} imposes the verb_type of the first verb in the VerbalForm, and the VerbalForm in the rule (the recursion) imposes the value it needs. It models, for example, *go, have, have gone, have gone to sleep, etc.* If we add the rules for other periphrasis in a similar way (the modal verb plus a VerbalForm with the adequate parameters), we may be able to express, for example: *I might have gone to have dinner before, I have been trying to find you, we have been thinking about going out to have dinner.*

The imposition of feature values and agreement makes it possible to write recursive rules sets, controlling the activation of each rule with the concrete values of the features of each particular word used.

All the improvements in the parser capabilities have increased the descriptive power, allowing us to model the linguistic phenomena we have considered more relevant for the formal grammar.

3.3.3.3.5 Pruning techniques

To reduce the time needed in the process due to the great amount of rules and the words ambiguity, a pruning algorithm using beam search has been included. It is a technique similar to the one described in [Coll96] with some modifications: the maximum probability rule is found, and it is used to set the threshold. All the rules with probability higher than the threshold will be expanded, and the rules with probability smaller will be eliminated. A variation has been included in the algorithm, and the branches are only pruned when the whole number of branches is higher than a certain number. Both thresholds have been experimentally selected, to reduce the time required to analyze, controlling that there was not a significant loose in the performance of the system.

3.3.3.4 Rewrite rules

Rewrite rules are the main information sources when using the prediction method based on the parser. They express the structures that will be considered grammatically correct. They were obtained from the study of text corpora, detecting the most frequent patterns, and were expressed as a sequence of *pos* adding also probabilistic information, features, lemmas, words and optional elements.

3.3.3.5 Error management

In the prediction method based on the parser, in each point of the analysis, the active rules are those that correspond with the left part of the sentence. When new words are written, the rules that do not expect their categories are eliminated, and the predictions proceed from the still active rules. If, after a sequence of words, all the rules are eliminated, the analysis would be stopped, and the prediction method would not be able to predict any *pos.*

The typology of parsing errors in our context free grammar is wider than in conventional grammars, due to its particular characteristics. The rules can fail for the following reasons: an unknown word (spelling mistake, technical word, etc.), lack of grammatical coverage, insertion, deletion or substitution of the expected *pos,* non valid features (they do not agree with the corresponding values, or they include a prohibited value, or not include an imposed value), the written word or its lemma do not coincide with the lemma or word imposed, the written word or its lemma coincide with the lemma or word prohibited.

To reduce the negative effects of these errors on the prediction process it is necessary to include error management algorithms to increase the flexibility. The current approach only handles the unknown words, using the unknown words categorization strategy explained in the section 3.3.2.1.4 (page 23). In case other error happens in the analysis, and it is stopped, there is a fall back to less powerful models, tripos or bipos and unigrams.

In the future lines chapter, some other possible error management techniques are outlined.

3.3.4 Additional techniques

In this sections, several auxiliary methods are presented. These techniques do not generate a prediction list, but they increase the prediction performance.

3.3.4.1 Elimination of rejected words

This method keeps a list of all the words that have been shown in the menu, and how many times. In the basic prediction methods, each word will appear until the user types a letter that did not match the word. If the user activates this filter, each word will be shown only once or twice. If a word is not selected, it is assumed that the user has seen it, and ignored it because it is not the desired one, so it will not be shown again, freeing positions in the predicted words list so that new ones can be presented to the user. If the desired word is in the main lexicon, fewer keystrokes will be needed for it to appear.

The results of this filter are very good, although it is better to *smooth* it, letting each word to appear more than once. Otherwise the user should type a word completely if the word appears in the menu only once and the user does not see it. The number of times the word should appear in the menu depends on several factors: the user's attention, his/her tiredness, his/her speed reading and writing, etc.

3.3.4.2 Endings prediction

As well as words, endings are also predicted (this is especially important for some words which are not included in the dictionary but have some typical endings).

3.3.4.3 Automatic character insertion

The system automatically inserts some characters in the text when punctuation marks are written. After typing a punctuation mark, the space before it (if it exists) is deleted, the character is inserted, and an extra white space is included after it. In case the character is a full stop, the next letter will be automatically capitalized.

3.4 Management module

The management module coordinates the information from the different prediction methods and dictionaries. It has allowed us to validate different integration methodologies in order to find the best combination. This module is in charge of:

- Obtaining and processing the input from the user interface (written text).
- Managing the information flow of each prediction method, coordinating the information they need and provide. Word prediction methods based on bipos or tripos require the *pos* of the previous word or words, and the features of the last word, and the parser needs all the grammatical information of all the words in

the sentence that it is been written. All of them provide a list of constraints for the following word.

- Managing the transactions with the lexicons. There are two different processes to manage:
  - The categorization: the main lexicon provides the management module all the information about a particular word.
  - The prediction: with different processes, depending on the dictionary:
    - The main lexicon provides words that follow the restrictions imposed by the prediction methods.
    - Personal and subject lexicons provide two different lists of words (each one). The first list contains the words that have followed a particular word in the training texts (word bigrams), and the second list contains the set of words in them that meet the constraints of the prediction methods.

  Of course, all the words in the lists start with the already written current word letters.
- Obtaining the list of predicted words from each method and lexicon, and selecting the definite set of predicted words. We have found that the optimum process to combine all the information available is:
  - First, to obtain the lists of bigrams of the last words, from the subject lexicon (if there is one active). If this list is not long enough, the list of bigrams from the personal dictionary is added.
  - In case the list is still not long enough, the management module gets the list of constraints of the more powerful grammatical prediction method: tripos (as it will be seen in the results chapter, the parser is still not so powerful as the tripos based method). In case the tripos does not predict a list of constraints, the bipos method will be used.
  - If a subject lexicon is active, the list of words from the subject lexicon that meet the constraints are included in the list, in order of frequency.
  - If the list is not long enough, the management module combines words from the main lexicon and the personal dictionary, pondering their probabilities according to the following formula:

$$p(w) = Q_{personal.} \cdot p_{personal}(w) + Q_{main} \cdot p_{main}(w)$$

Equation 4. Combination of the probabilities of the word in the main and personal lexicon.

Where:

- $p_{personal}(w)$ and $p_{main}(w)$ are the results of applying the Equation 1. Probability of the word w_t in the bipos model. extended for the tripos model.
- $Q_{personal.} = 0.35$ and $Q_{main} = 0.65$ are the weights that control the contribution of each dictionary. They have been trained to maximize the

number of keystrokes saved in texts between 1000 and 2500 words (the length of the text that a user may generate in a single session). If $Q_{personal.}$ is increased with respect to Q_{main}, the efficiency in smaller texts diminish, because the personal dictionary is still not enough trained. If $Q_{personal.}$ is reduced, the adaptation capacity decreases, and the results get worse, specially for long texts.

- If the list is still not complete, the more frequent words in the subject, personal and main lexicon (in that order) are added to the list, although they do not match the grammatical restrictions.
- In each one of the previous steps, the following verifications are made:
  - Whether the word has been shown to the user before (and rejected).
  - Whether the word is included or not in the main lexicon (in case it is necessary to control spelling mistakes).
- The list of endings that start with the last written letter is added to the word list.

- Managing the auxiliary methods.
- Sending the words and suffixes to the user interface.

3.5 User interface

The user interface is one of the critical parts in an application, determining the access of the user to the program. All the interfaces must be designed to be user-friendly, specially the ones devoted to people with disabilities.

Predice is a text editor specially designed to be used for people with physical disabilities, that can be used with the conventional keyboard, mouse, joystick, and one or two switches. The word prediction explained in this thesis is included in it. It has allowed us to test its usefulness of the prediction for real users, and see the way it can help them.

It has also been adapted for people with hearing and vision problems. It is highly customizable, so that people with very different needs and preferences can find a proper configuration that meets their requirements.

Special care has been taken in the design of the interface when it is used with a switch, to minimize the time needed to write a text with the scanning input method.

Adaptability mechanisms have been included in this interface, so that the program adapts its interface to the use done by each user, changing its configuration to optimize the access. The adaptability mechanisms included in *Predice* are:

- Position adaptability: the more frequently used options are moved to quicker positions, to accelerate its access.
- The options that are not used are disabled, and are not included in the scanning until the user activates them again. It has been taken into account that some

options can not be disabled, unless they are not used, such as some uncommon letters, and the options to escape in case of error, etc.

- Automatic adaptation of the scanning speed, considering the number of errors of the user (counted as the number of times the delete option and the escape options are used).

The user validates all the changes in the configuration before they are done to avoid the inconvenience of undesired changes in the interface. The implementation details of the adaptability mechanisms included in *Predice* can be found in [Garc00].

3.6 Detailed architecture

In the next figure the detailed architecture proposed in this Phd. Thesis is shown, as a reference framework for the descriptions included in this document.

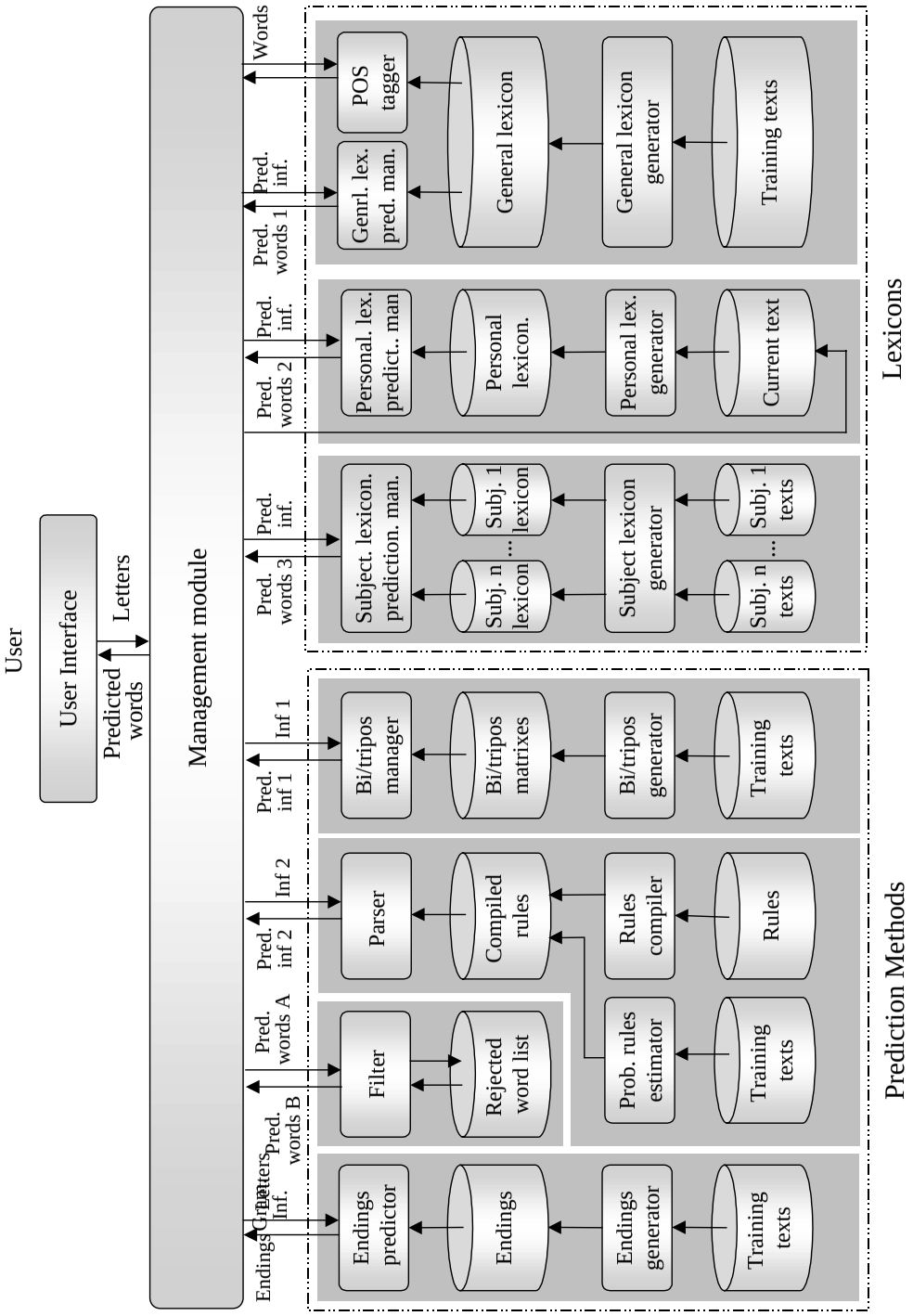

Figure 1. Detailed architecture of the word prediction system.

4 Evaluation of the word prediction

In this chapter, the efficiency of the prediction methods presented previously is evaluated, in order to compare them and select the optimum method or combination of methods. This chapter is divided into two parts: the first one, with the quantitative automatic evaluation, and the second one, with the user's subjective evaluation about the whole system, and some examples of user's text showing the benefits of the word prediction in the writing quality.

4.1 Automatic evaluation

In this section, a proposal is outlined for making automatic tests to obtain as much useful information as possible. On the one hand, it will be very interesting to have performance results valid for a comparison. On the other hand, knowing the best performance of the systems (with all the dictionaries, suffixes, grammatical information, extra features, etc.) will give the possibility to study the reasons for the differences in the results (or the lack of them). This will probably provide clues for future directions, showing the effects of changing prediction parameters, or using different methods, etc.

Before the description of the tests and its results, the decisions about the evaluation design, metrics and tests are justified. The usual criteria for NLP systems evaluation have been taken into account, adding a set of restrictions to consider the user's conditions in some tests.

4.1.1 Considerations about automatic evaluation

In the automatic evaluation of any system that includes natural language processing there are a factors that may affect the results, independently of the NLP system quality. For this reason it is necessary to control these factors, or at least take into account their effects, if comparisons between different methods is to be made.

4.1.1.1 Training and test corpora

The selection of the training and test corpora to be used in the system is one of the key factors to be considered: their careful design is essential, as system performance will depend heavily on them. In this section, some factors to take into account in the text selection are presented.

One of the parameters to decide upon is the *size* of the corpora. As in any NLP application, as much text as possible should be used, in order to obtain accurate and reliable models in the training stage, and statistically valid performance measurements in the tests. However, using large texts for evaluation can lead to unrealistic results. Take, for example, the case in which learning capabilities are included in the prediction. If very long texts are used with the adaptive prediction mechanisms, high levels of adaptation will be reached, producing good results. However, users will never write texts as long as those, so that the adaptation level reached in the evaluation will never be obtained and the performance in the real case will certainly be inferior. In automated laboratory tests it is possible, of course, to use as much data as is available.

The problem of using small corpora is that they may produce results highly dependent on the text and with little statistical significance, although this effect may be reduced if they are very carefully selected, with linguistic judgement. In case only a small corpus is available, it can be splitted into several parts. The algorithm is then trained with all the parts but one, and that one is used to test the system. The process is repeated several times, leaving out different parts, and finally averaging the results.

In general, strategies involving the execution of tests with several medium-sized texts are preferred, finally averaging the results.

In case of evaluation in *user's conditions*, training and testing corpora should be generated from texts entered by the user (logged data), totally adapted to his/her writing style. These are very difficult to obtain, especially large corpora for all the languages (except possibly for English) and confidentiality issues are usually involved. It should also be taken into account that logged texts will be over-adapted to a particular user. Further considerations on the evaluation of word prediction using logged data can be found in Copestake (1998).

Another key factor is the degree of *agreement* between the training and test sets: they must be completely different. Measurements taken with test corpora that overlap the training texts are not valid. Additionally, the limit should also be defined for considering the training and test corpora as being too closely related to lead to valid results. Great care must be taken when comparing different systems when they have been trained with different databases, considering the agreement between training and test material. If one of the systems is trained with data closely related to the application *domain* of the test material, it would outperform a system trained with data not so closely related (even when the latter uses better prediction methods). This is especially important when testing whole systems, in which the training corpora are not established.

4.1.1.2 Prediction features

In this section several prediction features which may influence the system performance to different degrees will be presented.

One of the more influential factors is the *number of predicted words* shown simultaneously to the user. Obviously, with more predicted words offered, less keystrokes are needed, because the desired word appears sooner, thus obtaining better performance results.

The maximum number of words shown is related to user considerations: in practice, too many words cannot be presented to the user at the same time. Previous works have established that most people can observe 5 words at a glance. This number keeps the right balance between extra cognitive load required and time saving: if more words are shown, the user needs more time and effort to read them when searching for the right one, making the prediction counter-productive, (unless the user decides to ignore it). Of course, for automatic word prediction evaluation in laboratory conditions any number of words can be used.

Some systems use suffix prediction when word prediction fails. This method is quite effective with new words, because it includes the most frequent endings. It should be decided whether this feature is included in the statistics, depending on the desired test: comparing systems with all their features, or only specific prediction methods. Regarding the *maximum number of suffixes* to set, the same considerations as with the maximum number of predicted words should be taken into account.

There are more techniques, which, strictly speaking, may not be considered prediction methods, but may accelerate the writing process when included in a system, thus influencing the performance results. For example, the *automatic inclusion of white space* and *autocapitalisation* of the first letter after a full stop reduce the number of keystrokes needed to write the text, or the *automatic elimination of the rejected words* (if a word appears in the menu several times and it is not selected, it will be assumed that it is not the desired one, and it will not be shown again while predicting the current word). As with suffixes, it should be decided whether these techniques are included in the evaluation or not.

It is also quite common that not only the prediction is evaluated in tests, but also the *interface* of the system into which it is integrated. For example, if the system has switch input with automatic scanning, the number of keystrokes are the number of times the user has to press the switch (two keystrokes/letter with row/column automatic scanning, one keystroke/letter with linear scanning, etc.).

The use of *accelerators* can lead to very different results: each time an accelerator is used, an extra keystroke is added to the statistics, reducing the savings in keystroke number, but saving time. With explicit rejection, no keystrokes are needed to select the predicted word. If the system is a window appearing on the screen, accessible with the normal keyboard, the keystrokes are from the keyboard, etc.

As stated before, the user interface aspect of system evaluation is not emphasized here. Nevertheless, it is a key point in word prediction applications evaluation, as the main

target of this research area is improving the communication skills of people with disabilities.

4.1.1.3 Metrics

In this section, certain measurements that seem to be relevant when evaluating word prediction performance are explained. Every parameter captures certain information about the prediction process, and the selected metric depends on the evaluator's interest.

The parameters that will be explained are keystroke savings, prediction coverage and learning rate. The use of time/speed measurements will be avoided because they are highly dependent on both user and interface design, and focus the description on measurements that can be automatically obtained from "standard" text corpora, not dependent upon consideration of the users.

4.1.1.3.1 Keystroke based measurements

In this subsection, measurements related to keystrokes will be described. This is one of the main points in word prediction systems evaluation, as these measurements are directly related to physical effort reductions from the user's perspective, especially important for people with physical disabilities.

There are two possibilities: to evaluate the keystrokes a user has to type or the keystroke savings due to the word prediction aid.

In the first case, the method is to measure the number of keystrokes a user has to type to enter the text (number of keystrokes before he/she is offered the desired word, plus the number of keystrokes needed to select the word). Depending on the level of detail required, the following measurements may be interesting:

- exhaustive graphs detailing the number of words versus the number of keystrokes needed to write them (the number of keystrokes typed before they are shown plus the keystrokes needed to select it, or the word length (measured in keystrokes, to be consistent) in case the word does not appear), or
- the average number of keystrokes needed to write a word, that is, in a more compact way. This figure could further be related to the average word length (measured in keystrokes) of the particular language (or corpus), for example, as a percentage, showing the average number of keystrokes typed per word.

The second case is complementary to the one already described. The method is to calculate how many keystrokes the user does not need to type. This can be considered more relevant from the user's perspective, because it actually represents the reduction in effort the user obtains when using the word prediction system.

The formula to calculate the percentage of keystroke savings is:

$$Savings(\%) = 100 * \frac{Keys_{woWP} - Keys_{WP}}{Keys_{woWP}}$$

Where:

- $Keys_{woWP}$ is the number of keystrokes needed to write the text without word prediction.

- $Keys_{WP}$ is the number of keystrokes needed to write the text using word prediction.

It is important to note that until now the measurements are based on keystrokes, but corpora are composed of characters. Therefore, some kind of relationship between keystrokes and characters needs to be derived, in order to apply objective keystroke measurements.

This relationship is dependent on the user interface and may not be the same in different systems, so the number of keystrokes per character should be standardized. For example, with a conventional keyboard, one character is equivalent to one keystroke (2 if the character is capitalized, or has an accent); in systems based on row-column matrices scanning, 2 keystrokes per character are needed, and 4 in the case of capital letters.

A unique equivalence list should be provided, because keyboard layouts vary according to the language. As a proposal, the number of keystrokes per character could be similar to the one using a conventional keyboard (with English keyboard layout):

1 keystroke per lower-case character (even if there is a tilde).

2 keystrokes per capital character (even if there is a tilde).

1 keystroke per number.

1 keystroke per sign which does not need to use the Shift key: , . ; -] # = \ / [\

2 keystrokes per sign which needs the Shift key: ! £ $ % ^ & * ( ) _ + { } @ < : | v ~ >

1 keystroke per selected word. If the system needs more than 1, it should be reported, because it can substantially affect the final result. It would also be very good to include both measurements.

4 keystrokes per other character.

Finally, it should be pointed out that some of the features mentioned in the section “Other system features” should also be considered when producing these figures, and must be explicitly indicated. For example, the final white space in the word should be included in the length, as it is automatically added when the word is selected, constituting an important part of the keystroke savings.

4.1.1.3.2 Prediction coverage measurements

In general, prediction coverage can be defined as the number of words correctly predicted, taking into account several factors that play a crucial role in the performance results. The idea is to evaluate to what extent the test corpus is covered by the prediction capabilities of the system.

These measurements are more important for dyslexic users, for example. Even when they are not able to write every word correctly, they may be able to recognize and select them in the menu, increasing their whole text quality, and probably quantity.

It is considered that a word has been correctly predicted if at least 1 character (and probably a white space) is saved. The measurement that is made here is the number or the percentage of words predicted in the text.

In case grammatical information in the training and test sets is available, similar measurements can be made for grammatical coverage, counting the number or percentage or categories correctly predicted, to evaluate the accuracy of the linguistic module.

Generally speaking, the following factors should be considered, as they influence the prediction coverage results:

- The number of *different* words in the training corpus.
- The number of *different* words in the test material.
- The language *perplexity or entropy*, because in languages with higher perplexity the prediction complexity is larger (for example, languages in which words have many different forms).
- The *grammatical coverage*. In case grammatical knowledge based on rules is applied, very different results may be obtained using texts with different levels of agreement with the rules. A metric for this grammatical coverage should be defined, possibly related to the percentage of sentences that follow the existing rules.
- The *agreement* between the test material and the system dictionaries, which can be measured as the number or percentage of words in the test texts that appear in the dictionaries. Note that it may be different from the number of words actually predicted due to two reasons:
  - if the system learns, each new word will be written only once, and subsequently it will be predicted
  - short words may be typed before they are predicted, even though they are in the dictionaries

4.1.1.3.3 Learning rate

Another important measurement is the learning rate, which consists of evaluating the learning speed of the prediction system as a measurement of the adaptability/flexibility of the prediction system. This is accomplished by computing the keystroke reduction at regular intervals throughout the input of a text, and representing a plot with the keystroke savings versus the number of words. In systems with learning capabilities, the keystroke saving improvement throughout the text can be observed. (Claypool, 1998).

4.1.1.3.4 Statistical validation

In a system where performance measurements are taken from a test text, the question of the possible generalization of the obtained conclusions raises. In the literature about statistics it is studied as statistical significance. A description for a general problem can be found in [Weis93], and there are specific references for statistical pattern recognition [Raud91], speech recognition [Gill89], natural language processing [Gibb98],

etc. There are several methods for statistical results validation, and in this PhD Thesis we use the confidence intervals, more widely used than other techniques, like the McNemar test [Gill89][Hunt90].

The formula to calculate the confidence interval is the following:

$$B_{/2} = \alpha \cdot \sqrt{\frac{\mathrm{Re}\,sult \cdot (1 - \mathrm{Re}\,sult)}{N}}$$

Where:

$B_{/2}$ is the length of half interval.

α is a parameter dependent on the confidence level (in this PhD Thesis, a confidence level of 95% has been chosen, and this sets α=1.96).

Result is the result of the experiment.

N is the number of elements in the test text: number of words in the text, of number of keystrokes needed to write it without prediction.

Once the length of the interval is calculated, the results of the experiment can be generalized for every text of the same domain, being sure the general results are inside the confidence interval:

$$\mathrm{Result}_{\mathrm{General}} = \mathrm{Results}_{\mathrm{experiment}} \pm B_{/2}$$

With the confidence level used in the formula. When different prediction methods must be compared, tests are performed applying each method to the same test text, and confidence intervals are computed for each one. A method is considered better than other one only if its result is better, and the intervals do not overlap. In other case, the conclusion is that the differences between them are not statistically significant with the chosen confidence level.

The confidence intervals size is inversely proportional to the test text length, so it can be shortened increasing the number of words in the text. Therefore, when results are obtained from small texts, bands may overlap, and the differences in the results may not be statistically significant. In case more text is available, test text length should be increased until intervals do not overlap (unless results converge when the number of words increases).

4.1.2 Description of the automatic evaluation system

The word prediction has been included in an evaluation system that allows the configuration of the parameters and the performance of different tests in an automatic way. This system is similar to the Parameterisable Test Bed (PTBs) described in Thompson (1994), considering the specific features of the word prediction. It is a tool that will automatically perform the whole evaluation process, not dependent on the user interface, flexible, and more general and complete than the already existing tools. We have attempted to make it as general as possible, although it may be used in a more reduced form in case there is an especially interesting subset (for example, text coverage evaluation using only

the 'X' dictionary and a particular prediction technique). The following steps should be carried out as automatically as possible for every single experiment (Gibbon 1998):

- Evaluation setup, in which the values of the parameters for the evaluation are established: language, training and test sets, maximum number of predicted words, number of keystrokes associated with each character, desired results, etc.
- Training procedures, in which the system is trained with the predefined training sets and the knowledge sources are extracted (dictionaries, rules, etc.).
- Test procedure: to perform the test under the established conditions.
- Scoring: generation of the results files according to the measurements to be done.
- Analysis: analysis of the results files, comparison between results of different experiments, or with the theoretical limits (maximum keystroke savings with a particular prediction method for a particular test, etc.)
- Reports generation. The reports which are generated would be complete enough to allow replication of the experiment, and extensive enough to help in the assessment process. For example, they should include at least the following information:
  - Values for the input parameters.
  - Values of other system features that may influence the results, (number of suffixes, etc.).
  - Parameters of the corpora: corpora length, average word length, percentage of common words, etc.
  - Results of each experiment.

It may be interesting to evaluate the actual performance limits. This involves the execution of at least two additional experiments:

- The first one, to determine the theoretical maximum performance for a particular prediction method. For example, to evaluate the maximum performance of grammatical methods, an experiment should be performed in which the part of speech of the following word is always the correct one.
- The second, to determine the lower limit for a particular test set which involves running the experiment without applying word prediction.

Sets of experiments have also been defined: each experiment was performed several times under different conditions: several test texts, with certain training texts, with the system's own dictionary, including and excluding particular features, etc. This set of experiments is exhaustive enough to allow an objective evaluation of the different methods and their combinations.

4.1.3 Results

In this section, the results of the experiments performed for each prediction method are shown. They are presented in series, with a first baseline experiment and a set of tests for different configurations in order to measure the effects produced by the change in the parameters of each technique.

For each experiment series, the following information will be provided:

- Description of the objectives.
- Name of the test text, number of words contained in it and number of keystrokes needed to write it without word prediction.
- Results of the baseline experiment: number of words predicted and keystrokes saved, both in absolute value and percentage, including confidence intervals for a confidence level of 95%.
- Discussion of the results.

4.1.3.1 Theoretical limit of the word prediction

In the following experiment, the theoretical limit of the word prediction is shown. The "prediction method" used consists in predicting always the desired word. This method predicts a 100% of the words, but it does not save all the keystrokes because, as the user has to select at least the predicted word and write the marks.

The text TESIS contains 37496 words and 253932 keystrokes are needed to write it without word prediction. A perfect word prediction method saves 191276 keystrokes (75.33%). If auxiliary mechanisms are used (autocapitalization, etc.), the number of keystrokes saved is increased up to 196295 (77.30%).

This is a theoretical maximum and the methods shown in this Ph.D. Thesis are far from this keystroke saving rate. However, it is important to consider that the maximum keystroke saving rate of the word prediction is not 100% but 75%.

4.1.3.2 Evaluation of the basic statistical methods

The basic prediction method, based only on the frequency of words, is now evaluated. This method depends only on the main dictionary, which is static, and no grammatical filtering is applied to its predictions.

In this experiment series, the influence of the number of words in the main lexicon is evaluated. In the experiments, the number of words of the main lexicon is limited to 200, 400, 800, 1200, 5000, 12000, 40000 and 132543 (the whole dictionary), always considering the most frequent words in the dictionary. The test text used is:

Test text name	TESIS
Number of words	37496
Keystrokes without word prediction	253932

The baseline experiment considers a main lexicon containing the 200 most frequent words in Spanish and its results are:

Number of predicted words	19211 (51.23%±0.51)
Number of saved keystrokes	36803 (14.49%±0.14)

The results of the experiments are:

# words in the main lexicon	# Predicted words	Relative improvement	# Saved keystrokes	Relative improvement
200	19211 (51.23%±0.51)	Baseline	36803 (14.49%±0.14)	Baseline
400	20574 (54.87%±0.50)	7.09%	42889 (16.89%±0.15)	16.54%
800	22870 (60.99%±0.49)	19.05%	52018 (20.49%±0.16)	41.34%
1200	23989 (63.98%±0.49)	24.87%	56527 (22.26%±0.16)	53.59%
5000	28408 (75.76%±0.43)	47.87%	74133 (29.19%±0.18)	101.43%
12000	31194 (83.19%±0.38)	62.38%	85127 (33.52%±0.18)	131.30%
40000	32682 (87.16%±0.34)	70.12%	90503 (35.64%±0.19)	145.91%
132543	33504 (89.35%±0.31)	74.38%	92367 (36.37%±0.19)	150.88%

As the lexicon size increases, there is always an improvement in the quality of the word prediction. The improvement is especially noticeable when working with small dictionaries. In all the experiments, the same words are predicted in the *zero letter,* independently of the previous words (the more frequent words in Castilian Spanish “de”, “la”, “en”, “que” y “en”), because no filter is applied. Improvements in the word prediction when the dictionary grows, are due to the increase in the predictions in the *letter two, three, four,* etc.

4.1.3.3 Evaluation of the statistical *pos* methods

In this series of experiments, the statistical *pos* models are evaluated. The effect of the basic *bipos* and *tripos* models on the prediction are shown, as well as the effect of the improvements included in them, and the results of the models trained from text tagged with ambiguity.

4.1.3.3.1 Evaluation of *bipos* model and influence of smoothing and fall back to unigrams

Bipos model prioritizes words belonging to the *pos* that more probably follow the *pos* of the last word. In these experiments, the same test text than in the previous series is used. The baseline experiment is the best one of the last series: using the whole main lexicon (130000 words).

Prediction method	# Predicted words	Relative improvement	# Saved keystrokes	Relative improvement
Whole main lexicon. Only unigrams.	33504 (89.35%±0.31)	Baseline	92367 (36.37%±0.19)	Baseline
Simple *bipos,* without smoothing or fall back	31050 (82.81%±0.38)	-7.32%	90437 (35.61%±0.19)	-2.09%
Bipos with floor smoothing	31053 (82.82%±0.38)	-7.32%	90533 (35.65%±0.19)	-1.99%
Bipos with fall back to unipos	32366 (82.32%±0.35)	-3.40%	90808 (36.94%±0.19)	1.56%
Bipos with fall back to unigrams	33232 (88.63%±0.32)	-0.81%	95742 (37.70%±0.19)	3.65%

In this table we can observe that the simplest bipos model works worst than the unigrams, in number of predicted words, and saved keystrokes. This result is due to the *number of zeros* in the matrix (the number of positions in the bipos matrix whose value is 0, because those *pos* sequences do not appear in the training texts). However, the percentage of words found in the *letter zero* increases from a 23.9% up to a 25.6%, so the quality of the word prediction in that letter increases. Alternative methods will be searched so that these improvements can be used, but always avoiding that the system global performance turns worse.

The smoothing of the *bipos* matrix with the *floor smoothing* method gives better results, comparing with those obtained from the simple bipos matrix. But they do not return better results than the unigrams do. Other well stablished and more sophisticated smoothing methods have been also applied (like the good-turing estimation) and obtain

results that are slightly better than the simple floor smoothing of the bipos matrix, but none of them obtains better performance than the method based on falling back to unigrams.

Next, fall back to several methods are tried. The first one uses a *bipos* matrix without smoothing, and fall back to *unipos* in case there are not enough predicted words. The number of predicted words in this experiment is still worse than the one produced by the unigrams method, but it increases the number of keystrokes saved, because predicted words are longer (its quality increases).

In the last experiment, *bipos* with fall back to unigrams are used. This is the best method based on *bipos,* with an improvement of 3.65% with respect to the baseline experiment. The number of predicted words is still smaller than in the baseline, but the keystroke savings has increased because of the improvement in the word prediction quality. In methods without grammatical filter, short words were always predicted (the first 5 words were always the same, and had 2-3 letters). The *bipos* information changes these words, and, as short words can only be shown in the *letter zero or one,* the user writes them before they are predicted, decreasing the number of predicted words.

4.1.3.3.2 Evaluation of *tripos* model and influence of smoothing and fall back to bipos and unigrams

Tripos model prioritizes words belonging to the *pos* that more probably follow the *pos* of the last 2 words. In these experiments, the same test text and baseline experiment than in the previous series are used.

Prediction method	# Predicted words	Relative improvement	# Saved keystrokes	Relative improvement
Whole main lexicon. Only unigrams.	33504 (89.35%±0.31)	Baseline	92367 (36.37%±0.19)	Baseline
Simple *tripos,* without smoothing or fall back	28436 (75.84%±0.43)	-15.13%	83374 (32.83%±0.18)	-9.74%
Tripos, with floor smoothing	29556 (78.82%±0.41)	-11.78%	85859 (33.81%±0.18)	-7.05%
Tripos with fall back to *bipos* and unigrams	33187 (88.51%±0.32)	-0.95%	96157 (37.87%±0.19)	4.10%

In the first experiments, can be observed a decrease in the performance of the system with respect to the baseline experiment. As the number of tagged words available is small (55000), the number of zeros in the *tripos* matrix, is very high (78.91%). The results of the method based on *tripos* with floor smoothing are similar to the ones obtained with *bipos* (with floor smoothing), still worst than the unigrams based prediction.

Different configurations, smoothing and fall back have been tried. The one with which better results are obtained is *tripos* with fall back to *bipos* and *unigrams* in case there

are not enough number of words in the prediction list, with a relative improvement of 4.10% in the number of keystrokes saved. The number of predicted words is smaller than in the baseline experiment because of the reasons already explained for the method based on *bipos*.

4.1.3.3.3 Evaluation of *bipos* and *tripos* trained from texts ambiguously tagged

In this series of experiments, *bipos* and *tripos* generated from text ambiguously tagged have been used.

The experiments tested in previous series (with and without fall back, with and without smoothing) have been tested here in the same way, returning similar results, only show the results of the best configurations are shown.

The same test text and baseline experiment than in the previous series are used.

In the first two experiments, training text is the same one used in the previous series (PAIS, 55000 words), being the only change the ambiguous categorization. The matrices used in the third test have been trained from a 1 million word text, not available without ambiguity.

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Whole main lexicon. Only unigrams.	33504 (89.35%±0.31)	Baseline	92367 (36.37%±0.19)	Baseline
Ambiguous *bipos* with fall back to unigrams (55000 words)	33244 (88.66%±0.32)	-0.78%	95210 (37.49%±0.19)	3.08%
Ambiguous *tripos* with fall back to ambiguous *bipos* and unigrams (55000 words)	33235 (88.64%±0.32)	-0.80%	95918 (37.77%±0.19)	3.84%
Ambiguous *tripos* with fall back to ambiguous *bipos* and unigrams (1 million words)	33319 (88.86%±0.32)	-0.55%	95951 (37.79%±0.19)	3.88%

As it can be observed, the number of saved keystrokes has been increased with respect to the baseline experiment. The results are not so good as the *bipos/tripos* trained without ambiguity, but the differences are not significant. It can be partially due to the fact that the ambiguity reduces the number of zeros in the matrices, reducing its negative effects. So, in case not enough tagged text is available, ambiguously tagged text can be used, returning satisfactory results (although not as good as the ones obtained from models generated from manually tagged text).

A more detailed study is needed before extrapolating these results to other *pos* models, with a higher number of categories, but the approach is promising, given there is no need to manually tag large corpora.

4.1.3.3.4 Evaluation of the effect of the categorization of the unknown words and the features management

In this series of experiments, the effect of two of the new features included in the basic *pos* models is evaluated. In one hand, the categorization of unknown words, as it has been explained in section 3.3.2.1.4 (page 23), and, on the other hand, the basic features management, performed in parallel with the prediction based on *bipos* and *tripos,* explained in section 3.3.2.1.1 (page 22).

The test text is the same one used in the previous experiment. The baseline experiment is the one that has obtained better results, the prediction based on *tripos*, trained from text tagged without ambiguity, with fall back to *bipos* and unigrams.

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Tripos with fall back to *bipos* and unigrams	33187 (88.51%±0.32)	Baseline	96157 (37.87%±0.19)	Baseline
Previous experiment including unknown words management	33165 (88.45%±0.32)	-0.07%	96252 (37.90%±0.19)	0.10%
Previous experiment including features management	33176 (88.48%±0.32)	-0.03%	98665 (38.85%±0.19)	2.59%

As it can be seen, when the unknown words management is included, the number of predicted words increases slightly (in a non significant number). The number of unknown words in the text is 3658 (9.76%). A 1.87% (700 words) are tagged using prefixes, endings, etc., and the rest 2958 words (7.89%) are assigned to the set of categories listed in section 3.3.2.1.4 (page 23).

In the last experiment, feature management is included, in parallel to *tripos,* making predictions matching in gender and number with the last written word. This information produces a significant improvement in keystrokes saving. The number of predicted words also decreases with respect to the baseline, because of the same reason: the predicted words are longer and save more keystrokes.

4.1.3.3.5 Evaluation of the prediction based on *pos* methods on newspaper texts

In this series of experiments, the best methods of each previous series are applied to a journal text with more than 100000 words, in order to check whether the tendencies observed are kept or depend on the text style. The test text used is:

Test text name	FEB100K

Number of words	103583
Keystrokes without word prediction	751191

This text is extracted from the electronic edition of the newspaper "El Mundo". This is the same newspaper used to train the main lexicon, so the style of this text is the style of the dictionary, (but the texts do not overlap).

The baseline experiment uses only information from the unigram model.

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Prediction based on unigrams	92012 (88.83%±0.19)	Baseline	266824 (35.52%±0.11)	Baseline
Bipos trained from text tagged without ambiguity with fall back to unigrams	91673 (88.50%±0.19)	-0.37%	273812 (36.45%±0.11)	2.62%
Tripos trained from text ambiguously tagged (1 million words), with fall back to bipos and unigrams	91820 (88.64%±0.19)	-0.21%	275479 (36.67%±0.11)	3.24%
Tripos trained from text tagged without ambiguity with fall back to *bipos* and unigrams	91514 (88.35%±0.20)	-0.54%	274192 (36.50%±0.11)	2.76%
Previous experiment including unknown words management	91473 (88.31%±0.20)	-0.59%	274195 (36.50%±0.11)	2.76%
Previous experiment including features management	91419 (88.26%±0.20)	-0.64%	280855 (37.39%±0.11)	5.26%

Most of the results could be expected, because they are similar to the results of the previous series (f. e., the effect of the features management). However, there are some differences, for example, in this case, the model based on *tripos* trained from a text ambiguously tagged (1 million words from "El Mundo", that does not overlap with the test text) produces better results than tripos train from a (smaller) text tagged without ambiguity. This may be due to the fact that the negative effects of the ambiguity are compensated by the decrease in the number of zeros and by the adaptation to the style of the text. So, training using a great ammunt of ambiguously tagged text, seems to be a reasonable option, because the results obtained are not very different to the results of *tripos* based on manually tagged text (much smaller), when large quantities of tagged text are not available. The relevance of this conclusion is related to the great cost of the tagging process.

4.1.3.3.6 Evaluation of the methods based on *pos* with small text

All the *pos* methods have been tested with a set of test texts with lengths varying between 986 and 3000 words, that could be written by a user in a single session. The test text used have been extracted from different domains: a newspaper text, a section of this thesis, two short stories and two parts of "*El Quijote*".

With each text, two experiments have been run: the first one, using only frequency information (unigrams) and the second, with the best *tripos* method, generated from text tagged without ambiguity, with fall back to *bipos* and unigrams, features management and categorization of unknown words.

The absolute results of the tripos models vary between a 84.71% and a 91.26% of predicted words and a 32.54% and 41.52% of keystroke savings, depending on the text, with a relative improvement between 4.68% and 6.62% in the number of keystrokes saved.

4.1.3.4 Evaluation of the formal method

In this section, the results of the prediction based on the context free grammar are shown. The test text is the first part of TESIS (the first 17500 words):

Test text name	TESIS (17500)
Number of words	17500
Keystrokes without word prediction	122425

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Prediction based on unigrams	15990 (91.37%±0.42)	Baseline	46310 (37.83%±0.27)	Baseline
The best prediction based on *tripos*	15854 (90.59%±0.43)	-0.85%	49195 (40.18%±0.27)	6.23%
Prediction based on *formal grammar* with fall back to *tripos*[*]	15537 (88.78%±0.19)	-2.83%	48883 (39.93%±0.27)	5.56%

As it can be observed, the formal method produces a reduction in the percentage of keystrokes saved although the confidence intervals overlap, so it cannot be concluded that the differences are significant.

The results are worst because the grammar is not perfectly adapted to the text that is been written. It should also be considered that TESIS contains a great amount of references

[*] Experiments have shown that the parser is able to analyze most of the input text, so the results are mainly due to the parser action, not to the fall back to tripos.

(not included in the dictionaries), and very long sentences, that may interrupt the parser, because this phenomena are still not included in the grammar.

It can be observed that in the *letter zero,* the number of predicted words increases from 23.1 (baseline experiment) to a 25.8 with the prediction based on tripos, and a 25.9 with the formal method, slightly better than the others.

4.1.3.4.1 Evaluation of formal method on different style texts

In this two sets of experiments, the prediction based on the formal grammar has been compared with the one based on tripos, using two different test texts.

First set of experiments, with a short story text:

Test text name	CUENTO1
Number of words	986
Keystrokes without word prediction	6470

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Prediction based on unigrams	865 (87.73%±2.04)	Baseline	2006 (31.00%±1.12)	Baseline
Prediction based on *tripos* with fall back to *bipos* and unigrams	847 (85.90%±2.17)	-2.08%	2104 (32.52%±1.14)	4.89%
Prediction based on *formal grammar* with fall back to *tripos*	819 (83.06%±2.34)	-5.32%	2091 (32.32%±1.14)	4.24%

Second set of experiments:

Test text name	QUIJOTE (1400)
Number of words	1390
Keystrokes without word prediction	8223

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Prediction based on unigrams	1203 (86.55%±1.79)	Baseline	2602 (31.64%±1.01)	Baseline
Prediction based on *tripos* with fall back to *bipos* and unigrams	1185 (8525%±1.86)	-1.50%	2687 (32.68%±1.01)	3.27%
Prediction based on *formal grammar* with fall back to *tripos*	1170 (84.17%±1.92)	-2.74%	2668 (32.45%±1.01)	2.54%

As it can be observed, the prediction method based on the context free grammar does not improve the results with respect to the tripos based methods, partly because the grammar has been trained from different style texts. Anyway the results are similar: although the text is not adapted to the grammar, the grammatical guide improves the results with respect to the results without any grammatical filter.

The results over text without any relationship with the grammar induce to think that using a grammar adapted the performance will be higher.

4.1.3.4.2 Effects of the prediction based on a formal grammar on texts adapted to the grammar

As it has been shown in previous texts, the grammatical guide of the parser does not produce strictly better results than the method based on tripos when the grammar is not adapted to the text (although it improves the results with respect to the prediction without grammatical guidance).

In the following test, the text has been slightly modified to adapt to the grammar. The test text has been called TESIS2, and it is a modified version of the first 2000 words of TESIS. The changes have been:

- Elimination of the biobliographical references inserted in the text, so that the analysis is not stopped due to them.
- Elimination of the word *etc.* when is inserted in the middle of a sentence, because the algorithm that segments sentences considers the period as an end of a sentence and restarts the analysis.
- Elimination (rewriting) of text between parenthesis.
- Elimination of the lists or enumerations.
- Rewriting of the very long sentences (those longer than 50 words). The average sentence length in TESIS2 is 18.56 words.

Both texts have bee included as an appendix in the Spanish version of this thesis, so that the slight changes can be seen there.

The characteristics of the modified text are:

Test text name	TESIS2
Number of words	2058
Keystrokes without word prediction	13957

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Prediction based on unigrams	1924 (93.49%±1.07)	Baseline	5732 (41.07%±0.82)	Baseline
Tripos with fall back to *bipos*	1919	-0.26%	6127	6.89%

and unigrams	(93.25%±1.08)		(43.90%±0.82)	
Prediction based on *formal grammar* with fall back to *tripos, bipos* and *unigram*	1899 (92.27%±1.15)	-1.30%	6193 (44.37%±0.82)	8.04%

As it can be seen, the keystroke savings increase with respect to the model based on *tripos,* and the percentage of words in the *letters zero and one* also increases from a 23.7% and 22.4% in the prediction based on unigrams to 27.3% and 22% in the prediction based on tripos and to a 27.4% and 22.7% for the prediction based on the formal grammar. We cannot conclude that the results are statistically significant, because the short length of the text results in big confidence intervals, and they are overlapped. However, the tendency to improve the overall perfomance can be observed, when the grammar is more adapted to the style of the text that is being written.

4.1.3.4.3 Effects of the fall back to tripos when the analysis is interrupted

In this series of experiments, the effect of the fall back to tripos (and bipos and unigrams) when the analysis is interrupted is tested out. The analysis can be stopped when an unknown word is written, or when the structure of the sentence is not included in the grammar. The error management method is the one used in the previous experiments, and in this one we want to show the comparison with the management of errors based on falling back only to unigrams.

The test text is the same that was used in the previous series, and the baseline experiment uses the prediction based on unigrams.

Prediction method	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Prediction based on unigrams	1924 (93.49%±1.07)	Baseline	5732 (41.07%±0.82)	Baseline
Prediction based on *formal grammar* with fall back to *unigrams*	1902 (92.42%±1.14)	-1.14%	6022 (43.15%±0.82)	5.06%
Prediction based on *formal grammar* with fall back to *tripos, bipos* and *unigram*	1899 (92.27%±1.15)	-1.30%	6193 (44.37%±0.82)	8.04%

As it can be observed, the second experiment (prediction method based on parser, with fall back to unigrams) produces better results than the model based only on unigrams, but does not surpass the prediction based on *tripos* with fall back to *bipos* and unigrams, shown in the previous series of experiments.

The last experiment is the same of the previous series, to remember the results of the fall back to tripos and bipos and unigrams when the analysis is interrupted. As can be observed, this error handling mechanisms is more powerful and its results surpass the ones produced by the tripos based method (when the grammar is adapted to the text).

4.1.3.5 Evaluation of personal and subject lexicons

In this section, the influence of including in the prediction process information about the subject and user's style is evaluated.

4.1.3.5.1 Personal lexicon effects on different length texts

As explained in the section 3.2.2 (page 9) about the personal lexicon, it is generated while the text is written, so, it is better adapted at the end of the session.

In the experiments with texts of tens of thousands of words (like the ones used in the previous experiments) the personal lexicon gets a high degree of adaptation to the text style, producing very good results in keystroke savings and number of predicted words. These results can be a good help to evaluate the dictionary in a theoretical way, but they do not represent the help they actually produce to the user, because the texts they produce in a single session will be much shorter, and the adaptation of the lexicon will not be so good. That is the reason why in some experiments short texts are used to test the behavior of the personal dictionary in the user's conditions.

In the following experiment series, different length texts are used. They correspond to the first words of TESIS, from the beginning to the number of words indicated in parenthesis. All the experiments use prediction based on *tripos* with fall back to *bipos* and unigrams, features management, and unknown words categorization. In the baseline experiment it is applied only on the main lexicon, while in the other one it is applied to the personal and lexicon using the procedure shown in section 3.4 (page 38) about the management module.

Experiment 1

Test text name	TESIS (100)
Number of words	100
Keystrokes without word prediction	707

Dictionary used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	87 (87.00%±6.59)	Baseline	273 (38.61%±3.59)	Baseline
Personal lexicon	86 (86.00%±6.80)	-1.15%	278 (39.32%±3.60)	1.83%

It can be seen that with a 100 word text there is an improvement in the number of keystroke saved, although the number of predicted words is slightly worst. Obviously, with so short texts, the confidence intervals are big and thus overlapped, so, it can not be concluded that a method is strictly better than the other, but when this test is repeated using different text, the results obtained are similar.

Experiment 2

Test text name	TESIS (200)
Number of words	200
Keystrokes without word prediction	1425

Dictionary used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	179 (89.50%±4.25)	Baseline	576 (40.42%±2.55)	Baseline
Personal lexicon	178 (89.00%±4.34)	-0.56%	608 (42.67%±2.57)	5.56%

When writing a 200 word text, the personal lexicon is better trained, and it produces an improvement in the results. The number of predicted words increases, The number of words found in the *letter zero*, that goes from 22.5% to 28.5%, specially due to the effect of word bigrams.

In the following experiment, the effect of the adaptation of the personal lexicon with the length of the text is shown. Both, the number of keystrokes saved, and the number of predicted words increase. Although the graphs are not shown, the number of words found in the *letter zero* increases at least a 5% in all cases, and in *letter one* between 5% and 12%.

Experiment 3

Test text name	TESIS (500)
Number of words	500
Keystrokes without word prediction	3483

Dictionary used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	443 (88.60%±2.78)	Baseline	1443 (41.43%±1.63)	Baseline
Personal lexicon	450 (90.00%±2.63)	1.58%	1587 (45.56%±1.65)	9.98%

Experiment 4

Test text name	TESIS (1000)
Number of words	1000
Keystrokes without word prediction	6764

Dictionary used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	895 (89.50%±1.90)	Baseline	2793 (41.29%±1.17)	Baseline
Personal lexicon	914 (91.40%±1.74)	-2.12%	3135 (46.35%±1.19)	12.24%

Experiment 5

Test text name	TESIS (5000)
Number of words	500
Keystrokes without word prediction	34264

Dictionary used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	4459 (89.18%±0.86)	Baseline	13773 (40.20%±0.52)	Baseline
Personal lexicon	4613 (92.26%±0.74)	3.45%	16956 (49.49%±0.53)	23.11%

Experiment 6

Test text name	TESIS (10000)
Number of words	10000
Keystrokes without word prediction	68805

Dictionary used	# Predicted	Relative	# Saved	Relative

	words	improv.	keystrokes	improv.
Main lexicon	8867 (88.67%±0.62)	Baseline	27311 (39.69%±3.36)	Baseline
Personal lexicon	9232 (92.32%±0.52)	4.12%	34784 (50.55%±0.37)	27.36%

Experiment 7

Test text name	TESIS
Number of words	37496
Keystrokes without word prediction	253932

Dictionary used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	33176 (88.48%±0.32)	Baseline	98665 (38.85%±0.19)	Baseline
Personal lexicon	34617 (92.32%±0.27)	4.34%	130834 (51.52%±0.19)	32.61%

As it is shown, from the first experiment, with 100 words, the personal dictionary produces an improvement in the prediction. In bigger texts the adaptation of the dictionary makes it predict better, specially the words that are not included in the main lexicon. Word bigrams increases the number of predicted words in the *letters zero and one:* for long texts, more than the 50% of the words are predicted in these letters.

4.1.3.5.2 Effects of predicting or non-predicting new words of personal lexicon

In this experiment the effect of including or not the new words* in the personal lexicon is shown. As explained in the section 3.2.2.1 (page 20), in users with problems to write, this words may be spelling mistakes, so, control mechanisms have been included to avoid these words to be predicted.

The test text for this experiment is:

Test text name	TESIS
Number of words	37496
Keystrokes without word prediction	253932

* Words not included in the main lexicon.

In the experiments, it is used prediction based on *tripos* with fall back to *bipos* and *unigram*, and features management.

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	33176 (88.48%±0.32)	Baseline	98665 (38.85%±0.19)	Baseline
Personal lexicon without new words	32554 (86.82%±0.34)	-1.87%	118305 (46.59%±0.19)	19.91%
Personal lexicon predicting new words used more than five times	33814 (90.18%±0.30)	1.92%	126001 (49.62%±0.19)	27.71%
Personal lexicon predicting new words used two times or more	34581 (92.23%±0.27)	4.23%	130300 (51.31%±0.19)	32.06%
Personal lexicon predicting always new words	34617 (92.32%±0.27)	4.34%	130834 (51.52%±0.19)	32.61%

As it can be observed, the system performance always increases (even when the new words are never shown), because of the bigrams, and the adaptation of the frequencies of the words. Of course, if new words are predicted, the system works better (in laboratory tests), but it should be considered if it can be counter-productive for users with writing problems and frequent spelling mistakes.

4.1.3.5.3 Evaluation of personal lexicon on small texts

Now, the behavior of the personal lexicon on the texts used to test the main lexicon in the section 4.1.3.3.6 (page 58) is shown. Two tests have been made with each one: the first one using tripos with fall back to bipos and unigrams with features management on the main lexicon, and the second one using the same prediction methods on the main and personal lexicons.

Experiment 1

Test text name	CUENTO1
Number of words	986
Keystrokes without word prediction	6470

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	848 (86.00%±2.17)	Baseline	2121 (32.78%±1.14)	Baseline

Personal lexicon	859 (87.12%±2.09)	1.30%	2380 (36.79%±1.18)	12.21%

Experiment 2

Test text name	CUENTO2
Number of words	2672
Keystrokes without word prediction	14878

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	2341 (87.61%±1.25)	Baseline	5107 (34.33%±0.76)	Baseline
Personal lexicon	2416 (90.42%±1.12)	3.20%	6230 (41.87%±0.79)	21.99%

Experiment 3

Test text name	CUENTO3
Number of words	1416
Keystrokes without word prediction	8682

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	1231 (86.94%±1.76)	Baseline	3134 (36.10%±1.01)	Baseline
Personal lexicon	1265 (89.34%±1.61)	2.76%	3474 (40.01%±1.03)	10.85%

Experiment 4

Test text name	MOLINOS
Number of words	2980
Keystrokes without word prediction	17715

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	2545 (85.40%±1.27)	Baseline	5765 (32.54%±0.69)	Baseline
Personal lexicon	2655 (89.09%±1.12)	4.32%	6867 (38.76%±0.72)	19.12%

Experiment 5

Test text name	QUIJOTE (2500)
Number of words	2446
Keystrokes without word prediction	14660

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	2072 (84.71%±1.43)	Baseline	4776 (32.58%±0.76)	Baseline
Personal lexicon	2139 (87.45%±1.31)	3.23%	5503 (37.54%±0.78)	15.22%

Experiment 6

Test text name	FEB1k
Number of words	1286
Keystrokes without word prediction	8536

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	1160 (90.20%±1.62)	Baseline	3368 (39.46%±1.04)	Baseline
Personal lexicon	1179 (91.68%±1.51)	1.64%	3515 (41.18%±1.04)	4.36%

Experiment 7

Test text name	TESIS (2000)
Number of words	2059
Keystrokes without word prediction	14267

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
Main lexicon	1879 (91.26%±1.22)	Baseline	5923 (41.52%±0.81)	Baseline
Personal lexicon	1920 (93.25%±1.08)	2.18%	6904 (48.39%±0.82)	16.56%

As can be seen, the use of the personal lexicon makes the performance of the system improve in all the experiments. In longer texts, the adaptation produces better results, except for the text FEB1K, which is extracted from a newspaper. In this case, the improvements of the personal lexicon are smaller, due to the adaptation of the main lexicon to its vocabulary. The only adaptation of the personal lexicon in this case is the prediction of the bigrams module.

4.1.3.5.4 Effects of subject lexicon

As it has been shown, the best performance of the personal lexicon is produced with long texts. That is the reason why this lexicon is stored, as a *subject lexicon,* so that it can be used in future sessions when the user writes about the same subject. In the following experiments, the effect of the subject lexicons are presented.

4.1.3.5.4.1 Effects of the subject lexicon on a text of the same domain

In this experiment, the prediction method are used on a subject lexicon about the same subject than the text. The test text is:

Test text name	QUIJOTE (8000)
Number of words	7813
Keystrokes without word prediction	46010

The baseline experiment uses the prediction method based on tripos with fall back to bipos and unigrams, with features management, on the main and personal lexicons. The subject lexicon has been obtained from the text QUIJOTE15-13K that, of course, does not include QUIJOTE(8000).

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
General and personal lexicon	6933 (88.74%±0.70)	Baseline	17943 (39.00%±0.45)	Baseline
Adding subject lexicon from QUIJOTE15-13K	7223 (92.45%±0.59)	4.18%	19381 (42.12%±0.45)	8%

An important improvement can be observed in the percentage of keystrokes saved and in the number of predicted words. The number of words predicted in the *letter zero* increases from a 28% to a 33.6%, mainly due to the bigrams included in the subject lexicon, and the adapted frequencies, that are applied from the beginning of the session.

4.1.3.5.4.2 Effects on a text of different domain

In this experiment it is shown the importance of a careful selection of the subject lexicon, because of its counter-productive effects when the subject of the text does not agree with the one of the lexicon. The previous experiments have bee repeated, but a subject lexicon generated from the text TESIS is been used.

Lexicon used	# Predicted words	Relative improv.	# Saved keystrokes	Relative improv.
General and personal lexicon	6933 (88.74%±0.70)	Baseline	17943 (39.00%±0.45)	Baseline
Adding subject lexicon from TESIS	6923 (88.61%±0.70)	-0.14%	16473 (35.80%±0.44)	-8.19%

Although word bigrams are the more powerful prediction method, they are also the more sensitive to the agreement degree (style, vocabulary, etc.) between the test and training texts. This is the reason why the results of this experiment are worse than the baseline when using a lexicon that does not agree with the subject of the text.

4.1.3.5.5 Effects of the additional techniques

In this series of experiment, the effects of the additional techniques are shown. As it was explained previously, this mechanisms do not predict words, but they also produce an acceleration in the writing rate.

The test text is QUIJOTE(8000), and the baseline experiment uses the prediction method based on tripos with fall back to bipos and unigrams, with features management, on the main, personal and subject lexicons. The subject lexicon has been obtained from the text QUIJOTE15-13K. In all the experiments, the configuration of the previous experiment is kept, adding another technique.

Prediction method	# Predicted	Relative	# Saved	Relative

	words	improv.	keystrokes	improv.
Prediction based on *tripos* with fall back to *bipos* and *unigram* using *general, personal and subject lexicons.*	7223 (92.45%±0.59)	Baseline	19381 (42.12%±0.45)	Baseline
Adding *automatic writing* of blanks after punctuation signs and capital letter after period.	7223 (92.45%±0.59)	0.00%	20773 (45.15%±0.45)	7.18%
Adding *endings prediction.*	7333 (93.86%±0.53)	1.52%	21106 (45.87%±0.46)	8.90%
Adding the filter to eliminate the rejected words that have appeared more than twice	7352 (94.10%±0.52)	1.52%	21249 (46.18%±0.46)	9.64%
Adding the filter to eliminate the rejected words that have appeared more than once	7401 (94.73%±0.50)	2.46%	21665 (47.09%±0.46)	11.78%

As it can be observed, all the additional techniques produce an improvement in the keystroke savings and/or the number of predicted words.

The best prediction method then includes auto-capitalization, automatic insertion of backspaces after punctuation marks, ending prediction and the filter to eliminate rejected words. This is the theoretically better configuration. However, it is recommended that this filter is not used strictly, but allowing each word to appear at least two times, (although the results are not so good), to avoid the penalty if the user does not recognizes the word he first time it appears in the menu.

4.2 Subjective evaluation

An intensive subjective evaluation of *Predice* or the word prediction has not been made, but we have reports that have been elaborated by psychologists and therapists about the use of Predice by users with different degrees of physical and mental disabilities. The general impression (of users and therapists) is very positive, agreeing in the advantages of the program and the word prediction in the communication, writing and learning processes. Some examples of user's text before and after using the program are available, showing the differences in the text quality (not only due to the use of the program, but also to the increase on the user's writing skills).

A short term evaluation has also been performed, to see the user's first impressions and the problems found by a person that is not used to the scanning interfaces of word prediction. The conclusion is that first session is hard, because it demands a great effort to adapt to the scanning, to find the letters and read the words, but after a certain time, the user considers the prediction very positive for switch users, and finds the program to be a very useful aid for people with physical disabilities.

It can be concluded that the use of the system, controlled in the learning stages, is positive in the medium to long term, producing an increase in the texts quality, and a decrease in the physical and cognitive effort needed to write them.

Users say that the use of the system makes to write text easier and quicker than other similar applications, and they so use word prediction (when the therapists allow it), so, they find it helpful and user friendly.

4.3 Comparison with other prediction systems

When evaluating word prediction systems, very different results may be obtained depending upon the test conditions and the actual implementation, because several factors not directly related to word prediction quality may influence them. In order to make valid performance comparisons, those factors should be controlled, or, at least, their effects taken into account (Palazuelos 1996). Some of the factors that may influence the results are:

- The differences among the languages. Differences may be found when doing a comparison of word prediction systems in several languages, due to the specific characteristics of each one. First of all, the effect produced by the same prediction method in each language may be different. In the second place, some of the methods applied for a particular language may not be feasible or may not make any sense in the others. For example:
  - The different behaviour of the verbs in English and Spanish: verbs in English have from 3 to 5 different forms, and, in Spanish, each verb (regular or irregular) has up to 53. This makes the prediction of verbs in Spanish very difficult, producing a large increase in dictionary size, or in the prediction algorithm complexity, depending on the method chosen to handle it.
  - In German, a very common method for generating new words is the concatenation of words. Some of the consequences in the word prediction mechanisms are: It makes words much longer, so many keystrokes are saved if one of them is predicted. The composed words can be predicted "component by component", which is impossible for other languages. It increases the training complexity, because a segmentation process is needed to determine the components to include in the dictionary. In case this is not possible and whole compound words are considered, the number of different words appearing in the texts is larger, the dictionary size increases, and also the probability that words are not included in it.
  - There are language-specific features, (not present in other idioms), that can help in the prediction process, such as morphologically encoded grammatical agreement between the article and noun in Spanish, which is not present in English.
- System specific features: automatic addition of white space, capitalisation of the first letter after a full stop, suffix prediction, etc.
- Differences in the training and test information, to test the performance of the word prediction methods in particular conditions.

- The results obtained: time or keystroke measurements, etc. and the method used to obtain the data necessary to calculate them.

For all the reasons mentioned above, it is very difficult to compare different systems, unless the test conditions are specially designed to allow a valid comparison. In [Pala98b], [Clay96], [Carl97a], [Stew96] and [Boek96] several results of different prediction systems are shown, indicating keystroke savings between 37 and 46%.

5 Conclusions

This Ph.D. thesis is aimed at the study of including linguistic information in word prediction for Spanish, with the main objective of improving the writing aids available for people with different kind of disabilities.

Word prediction involves a specific analysis, with its particular needs, which are different from generic studies about natural language processing. Other added value is the use of strategies that are specific to Spanish, taking into account that most of the available bibliography is about research or word prediction systems for English.

In order to include linguistic information, we propose a novel architecture that allows the development of an original methodology in order to combine the different sources of information we have explored (mainly in the lexical, morphological and syntactic levels), thanks to the inclusion of a management module, able to deal with and combine the different information flows used, and to the strict separation between the lexicons (main, custom and subject) and the prediction methods themselves. In every module including linguistic knowledge, we have made specific contributions, both in the design and organization of the information (mainly oriented to be used in the formal grammar) and in the particular methodology of using this information when facing word prediction and the adequate cooperation with other modules.

The prediction methods included use two main modeling strategies for the linguistic information: stochastic modeling (unigrams, bigrams, bipos and tripos) that considers a small amount of information of the written text (only the last words) and formal modeling (using a probabilistic context free grammar strengthened with additional characteristics).

An intensive evaluation has been performed, quantitative in the laboratory, and qualitative, with users evaluating a text editor with a word prediction system that follows the proposed architecture. The quantitative evaluation allowed us to analyze the capabilities of each source of information and choose the best combination. It is important to point out the lack of a standard (national or international) that determines which parameters should be evaluated, or the tests needed to do it. In this PhD Thesis we present a detailed study of the factors that influence the results and, therefore, should be standardized. We have also proposed the metrics to use, considering the ones that better evaluate the benefits of the prediction for each particular set of users. The qualitative evaluation means the gratification of knowing that the users like the results of our efforts.

In the following paragraphs we will describe in more detail the contributions of this PhD Thesis. We will start presenting the ones referred to the proposed architecture, the

organization of the information and the sources of knowledge used. After that, we will describe the contributions in the part of formal and stochastic *pos* modeling. Finally, we will sum up and discuss the results of the evaluation of the architecture implemented.

Once the different linguistic information sources have been studied and adapted, we have used their characteristics to build a global architecture that allows a novel combination of the information that we have considered optimum for the word prediction systems. In this architecture we propose a module for the management of the information flow that allows the integration of the power of each linguistic module (general, custom and subject lexicons, stochastic pos and formal models, rejected words filter, and endings predictor).

The main features of our architecture are:

- Its modularity, that allows the easy integration of new prediction methods and additional sources of knowledge, without the need of redesigning the architecture.
- Its flexibility, because the architecture is independent of the specific task (word prediction) and the language.
- Its power, for the capability of integrating multiple information flows.

In order to obtain a small amount of words using the sources of linguistic information, the words have been described in the dictionaries including, apart from the significant, its possible lemmas, parts of speech and features. This information has been carefully designed and organized for its use with the formal grammar. Probabilistic information has also been added to each word. We also consider a relevant contribution of this PhD Thesis the methodology of design and organization of this information, specially the final set of categories used and its description. The categories we propose are slightly different from the traditional ones, in order to better represent the syntactic behavior that can be observed in the texts, that we describe with the formal grammar we propose. The set of features is also different from the traditional one, as well as the original management mechanisms proposed, more powerful than the unification.

The capability to use information from the subject and custom lexicons (usually less trained than the main lexicon) has been possible thanks to the separation in different modules of the information linked to each word and the grammatical prediction models.

With respect to the formal model, the detailed study of linguistic phenomena (both theoretically and empirically) has led us to design a probabilistic context free grammar that uses an original interweaving of different mechanisms, that endow it with a significant descriptive power of the language:

- Powerful features management. As well as the concordance, it is also permitted to impose or prohibit a particular value for any feature of any of the terminal or non-terminal symbol included in each rule. This provides a great expressive power, allowing not only restricting the values (concordance, imposition or prohibition), but also including control mechanisms, e.g. the restriction of the set of rules to apply in a particular point of the analysis, depending on the value of the features of a non-terminal symbol.

- Grammar exception handling, with the possibility to impose or prohibit words or lemmas, in order to model the expressions where those words or lemmas have a particular behavior, different to the rest of the words of all the categories they belong to.
- Possibility of dealing with optional elements, allowing the writing of rules where the presence of a symbol is not mandatory.

The dictionaries have been carefully designed to include all the information required for the formal grammar. Both, the stochastic and formal models are based on parts of speech, to avoid the dependence with the individual words. Special care has been taken to decide the information of each word transferred from the categories to the features, so that the rules design is optimal.

As a sample of the power of the proposed grammar with the final set of categories and features, we have presented examples that would need more than 200 rules to be described with a traditional context free grammar, while in our proposal it is reduced to a single rule.

The information of the categories and features of each word has been also used for the stochastic model, allowing the description of *pos* models (bipos and tripos). These models have smaller training requirements than the ones based on words (word bigrams and trigrams). We have added to the *pos* models a filter to check the features, to partially compensate the loose of power due to the shift of part of the information from the categories to the features.

As the prediction methods are based on *pos*¸ part of our interest has focused on assigning the correct part of speech to the unknown words (words not included in the dictionary). Our proposal is a module that uses information of the word and the main lexicon to try to find out the right *pos* to assign to it. It uses a combination of the ending of the word, prefixes or suffixes, conjugation of the regular verbs, etc. In case this information is not enough, the module assigns a particular set of categories that has been trained, which is the more probable set of categories of the unknown words.

Our work is not only limited to a theoretical study. We have also implemented and evaluated a working system, built following the proposed architecture. This has allowed us the evaluation of each method, model and their combinations, and the selection of the optimal global prediction system. In this system, additionally, we have taken into account some considerations on the user interface design, which are: a good “generic” design, its adaptation to users with physical disabilities, and, finally, the inclusion of adaptive mechanisms, some of them specific for interfaces based on scanning.

With respect to the evaluation, from the study of the different factors that influence the word prediction, and how they do it, we propose a methodology for the evaluation. We propose two parameters to measure the quality of the word prediction: the percentage of predicted words, which is important for people with linguistic problems and the percentage of saved keystrokes, relevant for people with physical disabilities, for its direct relationship with the saved effort.

After the evaluation process, we propose the optimum combination of the information from the different dictionaries and prediction methods. In this combination, we prioritize words provided by the model based on word bigrams from the subject and

custom lexicons. After this, we use the stochastic *pos* models, applied first to the subject lexicon and afterwards, with an adequate weighting, to the custom and main lexicons. Of course, the words included in the list may start with the initial letters the user has typed. The weights to combine the information from the main and the custom lexicon have been optimized for small texts: the ones that can be generated in a single session for a person who needs a technical aid to write, because these are the actual conditions of use for the system.

With respect to the word prediction method based in the formal grammar, the overall set of contributions allowed us to get results close to those obtained with the stochastic pos models, leaving for future research the completion of its descriptive capabilities and a deeper analysis of the inclusion of other sources of knowledge, p. ej. including semantic information, to support the grammatical analysis, obtaining eventually significant improvements. The modularity and flexibility of the architecture will allow us to carry out this research work taking great advantage of the effort already invested here.

6 Future work lines

In the word prediction for Spanish applied to AAC there are still many possible fields to further research on, to establish future work lines. The main headlines are the following:

- Dictionaries:
  - Increase the flexibility of the main lexicon, including fragments of texts, etc.
  - Include new knowledge sources.
  - More powerful strategies for frequencies training.
  - More complex algorithm to combine the information from the different dictionaries, considering the training of each one, etc.
  - Automatic selection of subject lexicons, depending on the agreement between the text and the lexicon. Possibility to use several subject lexicons, pondered automatically.
- Formal grammar:
  - To increase the descriptive capacity of the formal grammar, to model a wider set of syntactic phenomena.
  - To refine the set of features and categories to support the new rules, manually by an expert, or automatically, considering automatic clustering techniques.
  - Inclusion of semantic information.
  - Increase the power of the parser and its robustness: new error management techniques.
  - New methods to train all the probabilities in the rules (probability of each rule, the optional symbols, etc.)
  - Flexibility in the rules application: dynamically modify the probability of each rule, depending on the detail level included in the rule (whether it is applied to a word, lemma or *pos).*
- Probabilistic pos models: use of longer *pos* sequences, or training with them neural networks, for example.
- Evaluation: it is necessary to establish a standard for the evaluation that allows the comparison between different systems. It should be different depending on

the evaluator's criterion (user with physical or linguistic disabilities, developer, etc.), measurements to evaluate the quality of texts, etc. A detailed user's model should also be made, to evaluate the impact of the different prediction methods in more realistic conditions.

- Applications of the word prediction: include it in other systems, such as speech recognition, other languages, use of other grammars, modify it to adapt to different problem, for example, aids for deaf people, etc.

7 References

[Acce00] Access Writer. http://snow.utoronto.ca/cgi/tad/showdev.cgi. Febrero de 2000.

[AENO98a] AENOR. Norma UNE 139801:1998 EX. "Informática para la salud. Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad de las plataformas informáticas. Soporte físico". 1998.

[AENO98b] AENOR. Norma UNE 139802:1998 EX. "Informática para la salud. Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad de las plataformas informáticas. Soporte lógico". 1998.

[Agui99] Aguilar, A. y Saumenll, C. "Adaptaciones del ordenador para facilitar la inclusión escolar de un alumno con parálisis cerebral de 12 años de edad". Comunicación y Pedagogía, 162. Páginas 55-58. 1999.

[Aïtm95] Aït-Mokhtar, S., Rodrigo-Mateos, J. L. "Segmentación de textos y análisis morfológico de textos en español con el sistema SMORPH". SEPLN nº 17. Páginas: 29-41. 1995.

[Alfo90] Alfonseca, M., Sancho, J., Martínez Orga, M. "Teoría de Lenguajes, Gramáticas y Autómatas". Universidad y Cultura. 1990.

[Alle94] Allen, J. "Natural language Understanding". Benjamin/Cummings Publishing Company, Inc. 2ª Ed. 1994.

[Auro99] Aurora. http://www2.edc.org/NCIP/library/wp/Aurora.html. Junio de 1999.

[Auro01] Aurora. http://www.djtech.com/Aurora/info/compare.html. Enero de 2001.

[Basi98] Basil, C. Soro-Carnats, E. y Rosell, E. "Sistemas de signos y ayudas técnicas para la comunicación aumentativa y la escritura: Principios teóricos y aplicaciones". 1ª edición. ISBN 844580716-1. Grupo Masson. 1998.

[Bate78] Bates, M. "The Theory and Practice of Augmented Transition Network Grammars". Natural Language Communication with Computers. Nueva York. Springer. Páginas: 191-259. 1978.

[Baum90] Baumgart, D., Johnson, J., and Helmesteter, E. "Augmentative and Alternative Communication Systems for persons with Moderate and Severe Dissabilities". Baltimore: Brookes. 1990.

[Berg00] Bergman, E., Johnson, E. "Towards Accessible Human-Computer Interaction". http://www.sun.com/tech/access/updt.HCI.advance.html. 2000.

[Bert95] Bertenstam, J., Hunnicutt, S. “Including Grammatical Information in word prediction” (Traducción de “Användning av grammatik y ordpredicering”). Research Conference on Människor-handikapp-livsvillkor (Man-handicap-conditions of life). Habiliteringsförvaltningen. Örebro. Suecia. 1995.

[Boek96] Boekestein, M. “MA Thesis: Word Prediction”. Dept. of Language and Speech. Katholieke Universiteit Nijmegen. Agosto de 1996.

[Bunt96] Bunt, H., Tomita, M. “Recent Advances in Parsing Tecnology”. Ed.: Kluwer Academic Publishers. 1996.

[Cald87] Calder, J. “Typed Unification for natural language processing” en Klein E. & J. van Benthem (eds) “Categories, Polymorphism and Unification”. 1987.

[Cald88] Calder J., Klein, E. & Zeevat H. “Unification Categorial Grammar: A Concise, Extendable Grammar For Natural Language Processing”. Proceedings of the 12 International Conference of Computational Linguistics and the 24 Annual Meeting of the Association for Computational Linguistics. Budapest. 1988.

[Cand97] Candelas Arnao, A., Lobato Galindo, M. “Guía de acceso al ordenador para personas con discapacidad”. Editado por el Ministerio de Trabajo y Asuntos Sociales. ISBN: 84-88986-71-8. 1997.

[Carl97a] Carlberger, J. “Design and Implementation of a Probabilistic Word Prediction Program”. Master Thesis.

http://www.speech.kth.se~johanc/thesis/thesis.html. Estocolmo (Suecia). 1997.

[Casa89] Casado, Enríquez, E.V. “The Spanish Category System”. Informe final Esprit-860. Vol I. Sección 1.2.1.1. UN-CAT1588. 1989.

[Cher94] Cherry, A., Hawley, M., Freeman, M., Cudd, P. “Human-Computer Interfacing for the Severely Physically disabled´´. Eds: Zagler, W.L., Busby, G. Wagner, R.R. “Computers for Handicapped Persons”. Proceedings of the 4 International Conference ICCHP'94. Vienna. Austria. Septiembre de 1994.

[Chur92] Church, G., Glennen, S. “The Handbook of Assistive Technology”. Chapman and Hall. Londres. 1992.

[Clar97] Clarkson, P. y Rosenfeld, R. “Statistical language modeling using the CMU-Cambridge toolkit”. Proceedings de EUROSPEECH'97. Páginas: 147-148. 1997

[Cola99] Colás, J. “Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión de habla continua en castellano”. Tesis Doctoral. ETSI Telecomunicación Universidad Politécnica de Madrid. 1999.

[Coll96] Collins, M. “A New Statistical Parser Based on Bigram Lexical Dependencies”. Proceedings de la 34ª Reunión anual de la Association for Computational Linguistics. 1996.

[Cope98] Copestake, A., Flickinger, D. “Evaluation of NLP technology for AAC using logged data”. ISAAC Research Symposium: Natural Language Processing and AAC. Dublin. 1998.

[CoWr00] CoWriter. Junio de 2000.

http://www.ndcd.org/ndcpd/people/staff/fifield/littech/tools/software/cowriter.htm.

[CoWr97] CoWriter. http://www2.edc.org/NCIP/library/wp/Cowriter.htm. Septiembre de 1997.

[Dema89] Demasco, P., McCoy, K., Gong, Y., Pennington, C., Rowe, C. "Towards more intelligent AAC interfaces: The use of natural language processing". RESNA Press. 1989. http://www.asel.udel.edu/nli/pubs/1989/DemaMcCo89.txt.

[Dema92] Demasco, P., McCoy, K. "Generating text from compressed input: An intelligent interface for people with severe motor impairments". Comms. of the ACM. Volumen 35. Nº 5.

http://www.asel.udel.edu/nli/pubs/1992/DemaMcCo92.txt. Mayo de 1992.

[Dema94] Demasco, P. "Human Factors Considerations in the Design of Language Interfaces in AAC". Assistive Technology. Volumen 6.1 Páginas: 10-25. 1994.

[Donn95] Donnelly, C., Stallman, R. "Bison. The YACC compatible Parser Generator. Bison Version 1.25".

http://www.gnu.org/manual/bison/html_mono/bison.html. Noviembre de 1995.

[Donn99] Donnelly, C., Stallman, R. "Bison. The YACC compatible Parser Generator. Bison Version 1.27".

http://www.delorie.com/gnu/docs/bison/bison_toc.html. 12 de Febrero de 1999.

[DRAE00] Diccionario de la Real Academia de la Lengua. 2000. http://www.rae.es/nivel1/buscon/AUTORIDAD2.HTM

[Eagl98] EAGLES Group. "Evaluation of Natural Language Processing Systems". Final Report. Septiembre de 1995.

[Earl70] Earley, J. "An Efficient Context-Free Parsing Algorithm". Comm. of the ACM. Volumen 13. Nº 2. Páginas: 94-102. Febrero de 1970.

[Esco99] Escoin, J. "PARLA y TaP: Programa de aceleración del rendimiento en lenguaje asistido y síntesis de voz en castellano". Comunicación y Pedagogía, 162. Páginas 67-73. 1999

[Ferr96] Ferreiros, J. "Aportación a los métodos de entrenamiento de Modelos de Markov para reconocimiento de habla continua". Tesis Doctoral. ETSI Telecomunicación Universidad Politécnica de Madrid. 1996

[Gara94a] Garay-Vitoria, N., González-Abascal, J. "Application of Artificial Intelligence Methods in a Word-Prediction Aid". Eds: Zagler, W.L., Busby, G. Wagner, R.R. "Computers for Handicapped Persons". Proceedings of the 4 International Conference ICCHP'94. Vienna. Austria. Septiembre de 1994.

[Gara94b] Garay-Vitoria, N., González-Abascal, J. "Using statistical and syntatic information in word prediction for input speed enhancement". Basque international workshop on information technology. BIWIT. Biarritz. Francia. 1994.

[Garc00] García Hernández, A. "Implementación de mecanismos adaptativos en la interfaz de usuario de un editor de texto para discapacitados motrices". Proyecto Fin de Carrera. Tutora: Sira E. Palazuelos Cagigas. E.T.S.I. de Telecomunicación. Dpto. de Ingeniería Electrónica. Septiembre de 2000.

[Gibb98] Gibbon, D., Moore, R. & Winsky, R. Eds. "Spoken Language Systems Assessment" (Part of the "Handbook of Standards and Resources for Spoken Language Systems". Volume III). Mouton de Gruyter. Berlin. 1998.

[Gill89] Gillick, L., Coz., S. "Some Statistical Issues in the Comparison of Speech Recognition Algorithms". IEEE International Conference on Acoustics. Speech and Signal Processing. Vol. S1, pp. 532-535. 1989.

[Gome94] Gómez, J. M., Goñi, J. M., González., J. C. "Un analizador sintácticopara gramáticas asociativas por la izquierda". Actas del X Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural. SEPLN 94. Córdoba. Julio de 1994.

[Goñi97] Goñi, J.M., González, J.C. y Montero, A. "ARIES: A lexical platform for engineering Spanish processing tools". Natural Language Engineering 3(4). Páginas: 317-345. Cambridge University Press. 1997.

[Gram96] "Gramática del español correcto". Santillana. ISBN: 84-294-5079-3. 1996.

[Haus87] Hausser, R. "Left-associative Grammar. Theory and implementation". CMU-CMT-87-104. 3 de Junio de 1987.

[Hunn85] Hunnicutt, S. "Lexical Prediction for a Text-to-Speech System". STL-QPSR 2-3/1985. KTH. Estocolmo. Páginas: 47-55. 1985.

[Hunn86] Hunnicutt, S., Neovius, L. "A lexical prediction system". Proceedings of the ICASSP 86. TOKYO. 1986.

[Hunn87] Hunnicutt, S. "Input and Output Alternatives in Word Prediction". STL-QPSR 2-3/1987. KTH. Estocolmo. Páginas: 15-29. 1987.

[Hunn89] Hunnicutt, S. "Using syntactic and Semantic Information in a Word Prediction Aid". Proceedings of the Eurospeech 89. Volumen 2. Paris. Páginas: 191-193. 1989.

[Hunt90] Hunt, M. J. "Figures of Merit for Assesing Connected-Word Recognisers". Speech Communication. Volumen 9. Páginas: 329-336. 1990.

[IBM00] IBM. Software Accessibility.

http://www-3.ibm.com/able/accesssoftware.html. 2000.

[Jeli91] Jelinek, F. "Self-organized language modeling for speech recognition". Páginas: 450-506. 1990. En Waibel. A., Lee, K. (eds): "Reading in Speech Recognition". Morgan Kaufman Publishers. 1991.

[Jeli99] Jelinek, F. y Chelba, C. "Putting Language Into Language Modeling". Keynote speech 1 en EUROSPEECH'99. Página KN-1. 1999

[Katz87] Katz, S. “Estimation of probabilities from sparse data for the language model component of a speech recognizer”. IEEE Transactions on Acoustics, Speech and Signal Processing, Marzo 1987, nº 35, pp. 400-401.

[Kong95] Kong Joo Lee et all. “A Robust Parser Based on Syntactic Information”. Proceedings of the 7 Conference of the EACL. Dublin. Ireland. Páginas: 223-228. 1995.

[Krul91] Krulee, G. K. “Computer processing of natural language”. Prentice Hall Inc. 1991.

[Kugl89] Kugler, “Unification of the word classes of the ESPRIT Project 860”. Informe final Esprit-860. Vol I. Sección 1.2.1.2. BU-WKL-0376. 1989.

[Lang97] Langley, P. “Machine Learning for Adaptive User Interfaces”. Proceedings of the 21 German Annual Conference on Artificial Intelligence. Páginas: 53-62. Freiburg. Alemania. Springer. http://www.isle.org~langley/adapt.html. 1997.

[Lang98] Langley, P., Fehling, M. “The Experimental Study of Adaptive User Interface”. Technical Report 98-3. Institute for the Study and Expertise. Palo Alto. CA. http://www.isle.org~langley/adapt.html. 1998.

[Lang99a] Langley, P. “User Modeling in Adaptive Interfaces”. Proceedings of the 7 International Conference on User Modeling. Banff. Alberta. Springer. Páginas 357-370. http://www.isle.org~langley/adapt.html. 1999.

[Lang99b] Langer, S., Hunnicutt, S., and Hickey, M. “Languaje processing techniques and resources for communication aids”. En “Augmentative and Alternative Communication: New Directions in Research and Practice”. Eds: Filip T. Loncke, John Clibbens, Helen H. Arvidson, Lyle L. Lloyd. ISBN: 1 86156 143 1. Páginas: 77-83. Editorial: Whurr Publishers. Londres. 1999.

[Lawr98] Lawrence, P. R. “EZ Keys For Windows Uniquely Satisfies AAC and Other Assistive Technology Needs”. CSUN 98.

http://www.dinf.ch/csun_98/csun98_141.htm. 1998.

[Lehm96] Lehmann, S. Oepen, S., “TSNLP Test Suites for Natural Language Processing” in COLING-96. Proceedings of the 16 International Conference on Computational Linguistics. Copenhague. Dinamarca. 5-9 de Agosto de 1996.

[Lesh98] Lesher, G. W., Moulton, B. J., Higgimbotham, J., “Techniques for Augmenting Scanning Communication”. AAC Augmentative and Alternative Communication. Volume 14. Páginas: 81-101. Junio de 1998.

[Lowe80] Lowerre, B. y Reddy R. “The HARPY Speech Understanding System”. En “Trends in Speech Recognition”, W. Lea, editor. Páginas 340-360. Prentice Hall. 1980.

[Mage91] Magerman, D. y Marcus, M. “Pearl: A Probabilistic Chart Parser”. Proceedings de la European ACL Conference. 1991.

[Magn97a] Magnuson, T. “Word Prediction as Linguistic Support for Individuals with Reading and Writing Difficulties”. En "The European Context for Assistive

Technology". Ed: Placencia Porrero, I., Puig de la Bellacasa, R. IOS Press, Ohmsha. Proceedings of the 2 TIDE Congress. París. 26-28 de Abril de 1995.

[Magn97b] Magnuson, T. "Profet II, a New Generation of Word Prediction: An Evaluation Study" Proceedings of the AAATE Conference. ISBN: 90 5199 361 7. Páginas: 153-157. Ed. IOS Press(George Anogianakis Et All). Tesalónica. Grecia. 29 Septiembre - 2 Octubre de 1997.

[Magn98] Magnuson, T. "Linguistic evaluation of Profet II: a pilot project". Proceedings ISAAC Dublin. Páginas: 479-480. Ed: Ashfield Publications. 1998.

[Mart96] Martin, S. "Effective Visual Communication for Graphical User Interfaces". www.cs.wpi.edu~matt/courses/cs563/talks/smartin/int_design.html. Junio de 1996.

[Mart99] Martin, S., Hamacher, C., Liermann. J., Wessel, F., Ney. H. "Assessment of Smoothing Methods and Complex Stochastic Language Modeling". Proceedings on CD-ROM of the Eurospeech 99. Volumen 5. Páginas: 1939-1942. Budapest (Hungría). 1999.

[Meri98] Merino Torres, F. "Implementación de una herramienta para procesamiento de lenguaje natural en entorno Windows". Proyecto Fin de Carrera. Tutora: Sira E. Palazuelos Cagigas. Departamento de Ingeniería de Circuitos y Sistemas de la E.U.I.T. de Telecomunicación. Octubre 1998.

[McCo95] McCoy, K. et al. "Some applications of Natural Language Processing to the Field of Augmentative and Alternative Communication". Technical Report. Computer and Information Sciences Department And Applied Science and Engineering Laboratories. University of Delaware/A.I. duPont Institute. Newark. DE 19617.

Http://www.asel.udel.edu/nli/pubs/1995/McCoDema95.txt

[Micr99] "Microsoft Windows Guidelines for Accessible Software Design". Diciembre de 1999.

http://www.microsoft.com/enable/dev/guidelines/software.htm.

[Mill56] Miller, G. A. "The magic number seven, plus or minus two: some constraints on our capacity for processing information." Psychological Review. Volumen 63. Páginas: 81-97. 1956.

[Morr92] Morris, C., Newell, A. F., Booth, L., Rickets, I. W., Arnott, J. L. "Syntax PAL: A System to improve the Written Syntax of Language-Impaired Users". Assistive Technology. Volumen 4. Páginas: 51-59. 1992.

[Morr98] Morrison, A., Martin, A. "Evaluating the Effectiveness of Word Prediction". Proceedings of the ISAAC 98. Dublin. Irlanda. Páginas: 230-231. 24-27 de Agosto de 1998.

[Nada84] Nadas, A. "Estimation of probabilities in the language model of the IBM speech recognition system". IEEE Transactions on Acoustics, Speech and Signal Processing. Agosto 1984, nº 32, pp. 859-861.

[Newe92] Newell, A. F., Arnott, J. L., Booth, L., Beattie, W., Brophy, B., Rickets, I. W. "Effect of the PAL Word Prediction System on the Quality and Quantity of Text Generation". Augmentative and Alternative Communication. Volumen 8. Nº 4. (Decker Periodicals Inc. Ontario. Canada (ISSN 0743-4618)). Páginas: 304-311. Diciembre de 1992.

[Ney91] Ney, H., Essen, U. "On Smoothing Techniques for Bigram-Based Natural Language Modelling". Proceedings of ICASSP 91. Toronto. Páginas: 825-828. Mayo de 1991.

[Ney92] Ney, H., Mergel, D., Noll, A., and Paesler, A. "Data driven search organization for continuous speech recognition". IEEE Transactions on Signal Processing, volumen 40(2). Páginas 272-281. 1992.

[Ney94] Ney, H., Essen, U. y Kneser, R. "On Structuring Probabilistic Dependencies in Stochastic Language Modeling". Computer Speech and Language, volumen 8(1). Páginas: 1-28. 1994

[Nies98] Niesler T.R., Whittaker E.W.D. y Woodland P.C. "Comparison of Part-Of-Speech and Automatically Derived Category-Based Language Models for Speech Recognitio". Proceeding de ICASSP'98. 1998

[Ortm96] Ortmanns, S., Ney, H. y Eiden A. "Language-Model LookAhead for Large Vocabulary Speech Recognition". Proceedings ICSLP'96. 1996.

[PAL00] PAL. http://alpha.mic.dundee.ac.uk/acsd/research/pred.html. Febrero 2000.

[Pere91] Pereira, F. y Schabes, Y. "Inside-Outside Reestimation from Partially Bracketed Corpora". Proceedings de la 30ª Reunión anual de la Association for Computational Linguistics. Páginas 128-135. 1991.

[Pred99] PredictAb1ility. http://www.inclusive.co.uk/catalog/predict.htm. Septiembre de 1999.

[Rabi89] Rabiner, L. R. "A tutorial on hidden Markov models and selected applications in speech recognition". Proceedings of the IEEE, 77(2):257-286.

[RAE00] Real Academia de la Lengua. 2000.

http://www.rae.es/NIVEL1/CONSULTAS/DEMOSTRATIVOS.HTM

[Raud91] Raudys, S., Jain., A., "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 13, nº 3. Marzo 1991.

[Ricc96] Riccardi, G., Pieraccini, R., Bocchieri, E. "Stochastic automata for language modeling". Computer Speech and Language. 10. Páginas: 265-293. 1996.

[Rios99] Ríos, A. "La transcripción fonética automática del Diccionario Electrónico de Formas Simples Flexivas del español: estudio fonológico del léxico". Estudios de Lingüística Española 5. 1999.

[Rodr98] Rodríguez Paíno, L. "Diseño de un editor de textos predictivo para personas discapacitadas bajo entorno gráfico". Proyecto Fin de Carrera. Tutora: Sira E. Palazuelos Cagigas. E.T.S.I. de Telecomunicación. Dpto. de Ingeniería Electrónica. Julio 1998.

[Sanc99] Sánchez, J., Benedí, J., "Learning of stochastic context-free grammars by means of estimation algorithms". Proceedings of the Eurospeech 99. Volumen 4. Budapest (Hungría). Páginas: 1799-1802. 1999.

[Sani99] San Inocente Benito, F. "Actualización e Incorporación de Mejoras a un Editor Predictivo Orientado a Personas con Discapacidades para Entorno Windows". Proyecto Fin de Carrera. Tutora: Sira E. Palazuelos Cagigas. E.T.S.I. de Telecomunicación. Dpto. de Ingeniería Electrónica. Junio de 1999.

[SAW00] SAW. http://www.ace-centre.org.uk/html/sawalt1.html. Noviembre de 2000.

[Simp99] Simpson, R. C., Horstmann Koester, H. "Adaptive One-Switch Row-Column Scanning". IEEE Transactions on Rehabilitation Engineering. Volume 7. Número 4. Páginas: 464-473. Diciembre de 1999.

[Smit96] Smith, S. L., Mosier, J. N. "Guidelines for designing user interface software". http://www.syd.dit.csiro.au/hci/guidelines/sam/guidelines.html. The MITRE Corporation. Bedford. Massachusetts. E.E.U.U. Agosto de 1996.

[Soot00] SoothSayer Word Prediction. http://www.ahf-net.com/sooth.htm. Junio de 2000.

[Stei94] Steinbiss, V., Tran, B.H. y Ney, H. "Improvements in Beam Search". Proceedings de ICSLP'94. Páginas 2143-2146. 1994.

[Stol95] Stolcke, A. "An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities". Computational Linguistics. 21 (2). Páginas: 165-201. 1995.

[Suar98] Suárez, M.D., Aguilar, A., Rosell, C. y Basil, C. "Ayudas de alta tecnología para el acceso a la comunicación y la escritura". En Basil, C. Soro-Carnats, E. y Rosell, E. "Sistemas de signos y ayudas técnicas para la comunicación aumentativa y la escritura: Principios teóricos y aplicaciones". 1ª edición. ISBN 844580716-1. Grupo Masson. 1998.

[Subi00] Subirats Rüggeberg, C. y Ortega Gil, M. "Tratamiento automático de la información textual en español mediante bases de información lingüística y transductores". Estudios de Lingüística Española 10. 2000

[Suka92] Sukaviriya, P. N., Foley, J. D. "Built-in User Modelling Support, Adaptive interfaces, and adaptive help in UIDE". Graphics, Visualization and usability center. Georgia Institute of Atlanta. Octubre de 1992.

[Stew96] Stewart, H. "Just How Useful is Word Prediction?". ABILITY NEWSLETTER. Noviembre de 1996.

http://www.abilitycorp.com.au/news_views/newsletter_nov_1996.htm.

[Swif87a] Swiffin, A.L., Arnott, J.L, Pickering, Newell, A.F. "Adaptive and predictive Techniques in a communication Prosthesis". Augmentative and Alternative Communication. Volumen 3. Nº 4. Páginas: 181-191. Diciembre de 1987.

[Swif87b] Swiffin, A.L., Arnott, J.L, Newell, A.F. "The use of syntax in a predictive communication aid for the physically handicapped". Proceedings of the 10

Annual Conference on Rehabilitation Technology. San Jose. CA: RESNA. Páginas: 124-126. 1987.

[Thom94] Thompson, H. "TEMAA: A testbed study of evaluation methodologies: Authoring aids, Proceedings of the ELSNET Language Engineering Convention". Páginas: 147-148. Paris. 1994.

[Vall92] Vallés Botella, M. "Editor Comunicador Predictivo para Personas con Graves Limitaciones Motrices". Proyecto Fin de Carrera. Tutor: Francisco Giménez de los Galanes Cejudo. E.T.S.I. de Telecomunicación. Dpto. de Ingeniería Electrónica. 1992.

[Vand91] Van Dyke, J. "Word Prediction for Disabled Users: Applying Natural Language Processing to Enhance communication" BA Thesis. University of Delaware. 1991.

[Vila98] Vilaseca, D. "Daniel: el ordenador, una necesidad para el discapacitado, no un capricho". En Basil, C. Soro-Carnats, E. y Rosell, E. "Sistemas de signos y ayudas técnicas para la comunicación aumentativa y la escritura: Principios teóricos y aplicaciones". 1ª edición. ISBN 844580716-1. Grupo Masson. 1998.

[VOX00] Diccionario General de la Lengua Española VOX. 2000. http://www.vox.es/consultar.html.

[Weis93] Weiss, N.A. y Hasset, M.J. "Introductory Statistics" 3º edición, 1993.

[Witt91] Witten, I.H. y Bell, T.C. "The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression". IEEE Transactions on Information Theory, volumen 38(4). 1991.

[Wood96] Woods, W.A. "Augmented Transition Networks for Natural Language Analisys". Harvard Computation Laboratory Report No. CS-1. Cambridge. MA: Harvard University. 1996.

[Writ00] Write Away 2000. http://www.is-inc.com/product2.htm. Junio de 2000.

8 Author references

[Carl97b] Carlberger, A., Carlberger, J., Magnuson, T., Palazuelos-Cagigas, S.E., Hunnicutt, M.S. & Aguilera, S. “Profet, a New Generation of Word Prediction: An Evaluation Study”. Proceedings of the Workshop on NLP for Communication Aids, ACL/EACL'97. Páginas: 23-28. Ed. "Association for Computational Linguistics". Madrid. 1997.

[Clay98] Claypool T., Ricketts I., Gregor P., Booth L., Palazuelos S. “Learning Rates of a Tri-Gram Based Gaelic Word Predictor”. Proceedings ISAAC Dublin. 1998. Páginas: 178 - 179. Ed: Ashfield Publications. 1998.

[Pala94] Palazuelos, S.E. “Incorporación de mejoras ergonómicas y mecanismos predictivos a un editor orientado a personas discapacitadas”. Proyecto Fin de Carrera. Tutor: Juan Manuel Montero Martínez. E.T.S.I. de Telecomunicación. Dpto. de Ingeniería Electrónica. Noviembre de 1994.

[Pala95] Palazuelos, S.E., Montero, J.M., Gómez S., Aguilera S. “On the Development of a Word Processor With Word Prediction for Severely Physically Handicapped, Non-Vocal Users: PREDICE”. Proceedings of ECART 3. Páginas: 119-121. Lisboa. Octubre de 1995.

[Pala96] Palazuelos-Cagigas, S. E., Aguilera Navarro, S. “Report on Word Prediction for Spanish”. Informe WP7T3D.2IR del Proyecto Europeo “VAESS: Voices, Attitudes and Emotions in Speech Synthesis”. TIDE Nº 1174. Agosto de 1996.

[Pala97] Palazuelos-Cagigas, S. E., Godino-Llorente, J.I., Aguilera Navarro, S. “Comparison Between Adaptive and Non-Adaptive Word Prediction Methods in a Word Processor for Motorically Handicapped Non Vocal User” Proceedings of the AAATE Conference. ISBN: 90 5199 361 7. Páginas: 158-162. Ed. IOS Press(George Anogianakis Et All). Tesalónica. Grecia. 29 Septiembre - 2 Octubre de 1997.

[Pala98a] Palazuelos, S., Aguilera S., Rodrigo J., Godino J. “Grammatical and statistical word prediction system for Spanish integrated in an aid for people with disabilities”. ISBN: 1-876346-17-5. ICSLP'98. Sydney. 30 Noviembre - 4 Diciembre de 1998.

[Pala98b] Palazuelos, S., Aguilera, S., Claypool, T., Ricketts, I., Gregor, P. “Comparison of Two Word Prediction Systems Using Five European Languages”. Proceedings ISAAC. Dublin. Páginas: 192 - 193. ISBN: 1 897606 04 4. Ed: Ashfield Publications. 1998.

[Pala98c] Palazuelos, S., Aguilera, S., Ricketts, I., Gregor, P. y Claypool, T. "Artificial Neural Networks applied to Improving Linguistic Word Prediction". Proceedings ISAAC. Dublin. Páginas: 194-195. ISBN: 1 897606 04 4. Ed: Ashfield Publications. 1998.

[Pala99a] Palazuelos, S. E., Rodrigo, J. L., Godino, J. I., Aliaga, F., Martín, J. L., Aguilera, S. "Predicción de palabras en castellano". SEPLN Procesamiento del lenguaje natural. Revista nº 25. ISSN: 1135-5948. Páginas: 151 - 158. Septiembre de 1999.

[Pala99b] Palazuelos Cagigas S. E., Aguilera Navarro, S., Rodrigo Mateos, J. L., Godino Llorente,J. I., Martín Sánchez, J. L. "Considerations on the Automatic Evaluation of Word Prediction Systems". En "Augmentative and Alternative Communication: New Directions in Research and Practice". Eds: Filip T. Loncke, John Clibbens, Helen H. Arvidson, Lyle L. Lloyd. ISBN: 1 86156 143 1. Páginas: 92-104. Editorial: Whurr Publishers. Londres. 1999.

[Pala00] Palazuelos Cagigas S. E., Martín Sánchez, J. L., Godino Llorente, J.I., Rodrigo Mateos, J. L., Arenas García, J., Aguilera Navarro, S. "Estrategias de comunicación utilizando PredWin como comunicador". En: Libro de Actas del Congreso Iberoamericano Iberdiscap 2000. 3º de Comunicación Alternativa y Aumentativa, 1º de Tecnologías de Apoyo para la Discapacidad. Páginas: 277-280. ISBN: 84-699-3253-5. Madrid. 18-20 de Octubre de 2000.