Advances in NLP applied to Word Prediction

Carlo Aliprandi[1]*, Nicola Carmignani*[2]*, Nedjma Deha*[2]*, Paolo Mancarella*[2]*, Michele Rubino*[2]

[1]Synthema Srl – Pisa, Italy
[2]Department of Computer Science – University of Pisa, Italy
carlo.aliprandi@synthema.it, {nicola, deha, paolo, rubino}@di.unipi.it

Abstract

Presenting some recent advances in word prediction, a fluorishing research area in Natural Language Processing, we describe FastType, an innovative word prediction system that outclasses typical limitations of standard techniques when applied to inflected languages. FastType is based on combined statistical and rule-base methods relying on robust open-domain language resources, that have been refined to improve Keystroke Saving. Word prediction is particularly useful to minimise keystrokes for users with special needs, and to reduce misspellings for users having limited language proficiency. Word prediction can be effectively used in language learning, by suggesting correct words to non-native users. FastType has been tried out and evaluated in some test benchmarks, showing a relevant improvement in Keystroke Saving, which now reaches 51%, comparable to what achieved by word prediction methods for non-inflected languages.

Index Terms: Word Prediction, Natural Language Processing (NLP), Augmentative and Alternative Communication, Computer Aided Language Learning, Speech and Natural Language Interfaces, Assistive Technology

1. Introduction

This paper describes an innovative approach to Word Prediction, presenting recent results achieved for inflected languages.

Word Prediction is the task of guessing words that are likely to follow a given fragment of text. A Word Prediction software is a writing support: at each keystroke it suggests a list of meaningful predictions, amongst which the user can possibly identify the word he is willing to type. By selecting a word from the list, the software will automatically complete the word being written, thus saving keystrokes.

Word prediction is facing a very ambitious challenge, as several typical complex problems arising when dealing with Natural Language are to be faced. The inherent amount of arising ambiguities (lexical, structural and semantic ambiguities but also pragmatic, cultural and phonetic ambiguities for speech) are complex problems to be solved by a computer. Many research efforts have been experimented and several core NLP tasks have been employed as, for example, Language Modeling, Part-of-Speech (POS) Tagging, Parsing and Lemmatisation.

Word prediction has been widely adopted in Augmentative and Alternative Communication (AAC) systems [1], becoming an essential aid for people with motor or cognitive disabilities, in order to reduce the typing effort and to assist learning or language impairments. Indeed, writing text for work, study or communicating is, according to a survey we conducted (as described in [2]), the most frequent and time-consuming activity for most computer users. Therefore a word predictor would be useful to a very large number of computer users, both disabled and not.

FastType is designed to predict words for inflected languages, that is languages that have a large dictionary of word forms with several morphological features, produced from a root or lemma and a set of inflection rules. The degree of inflection of a language may vary from very high (e.g. Basque), to moderate (e.g. Spanish, Italian, French), to low (e.g. English). The large number of word forms makes word prediction for inflected languages a hard task. As word prediction operates at typing time, any NLP task that can be applied, unlike common NLP analytics which processes complete sentences, has to cope with the further problem of sentence incompleteness.

To make word prediction as simple and immediate as possible, we have implemented DonKey, a new human-computer interface. DonKey improves the original, naive, interface of FastType, allowing the user to benefit from automatic word prediction in any desktop application. In addition to re-designing the user interface, the underlying prediction engine has been enhanced: we added new resources, like the word and Part-of-Speech *n*-gram Language Models, and implemented more efficient prediction algorithms.

Thanks to the upgrades, performances are greatly improved. Keystroke Saving reached 51% and is now comparable to the one achieved with state-of-the-art methods for non-inflected languages.

2. State of the Art on Word Prediction

Word prediction is a research area where a very challenging and ambitious task is faced, basically with methods coming from Artificial Intelligence, Natural Language Processing and Machine Learning.

The main goal of word prediction is guessing and completing the word a user is willing to type. Word predictors are intended to support writing and are commonly used in combination with assistive devices such as keyboards, virtual keyboards, touchpads and pointing devices. Another potential application is in text-entry interfaces [3] for messaging on mobile phones and typing on handheld and ubiquitous devices (e.g. PDAs or smartphones).

Prediction methods have become quite known as largely adopted in mobile phones and PDAs, where *multitap* is the input method. Nuance T9 (formerly Tegic Communications T9)[1] and Zi Corporation eZiText[2] are commercial systems that adopt a very simple method of prediction based on *dictionary disambiguation*. At each user keystroke the system selects the letter between the ones associated with the key guessing it from a dic-

[1]http://www.nuance.com/t9/
[2]http://www.zicorp.com/eProducts/ZiPredictiveTextSuite/

tionary of words: hence they are commonly referred to as letter predictors. Letter predictors bring a Keystroke Saving (KS) but it has been proven to be not completely free from ambiguities that are more frequent for inflected languages. So it is not surprising that these methods had a great success for non inflected languages such as English: the limited number of inflectional forms lead to very high KS that, at the moment, are above 40%.

Word prediction is a more sophisticated technique within recent research. Differently from letter predictors, word predictors typically make use of language modelling techniques, namely stochastic models that are able to give context information in order to improve the prediction quality.

FASTY [4] is a statistically based adaptive word prediction program. The FASTY Language Model utilizes word unigrams, word bigrams, POS trigrams and the probability distribution $P(t|w)$, i.e. the probability that POS tag t occurs with a given word w.

Most of the literature related to word prediction concerns non-inflected languages [5]. In [6] and [7] Language Models and prediction techniques are presented that allow the user to save more than 50% of keystrokes. The contribution of the system presented in this paper is the adaptation and improvement of these techniques for inflected languages.

The language that the system has to model influences the prediction techniques; inflected languages pose a harder challenge to prediction algorithms, since they have to deal with a usually high number of inflected forms that dramatically decrease Keystroke Saving [8]. To simplify the task of predicting the correct form, some techniques [9] provide a two-step procedure, choosing first only among word "roots", and proposing all the possible word forms only when the user selects a root. FastType relies instead on Part-of-Speech (POS) and related morpho-syntactic information to provide a one-step procedure, presenting to the user a list of word forms. This procedure, combined with on-the-fly POS tagging, enables FastType to boost performances, cutting off of the prediction list all words whose gender, number, tense or mood are not consistent with the sentence context. The prediction list becomes also a "guide tool" to write syntactically correct sentences.

3. Description of the Word Predictor

Figure 1 shows the three main components of the FastType system: the *User Interface*, the *Prediction Engine* and the *Linguistic Resources*.

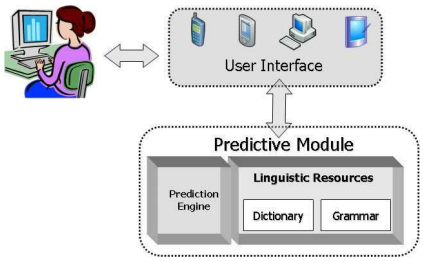

Figure 1: *FastType Architecture.*

The *Prediction Engine* is the kernel of the *Predictive Module* since it manages the communication with the User Interface, keeping trace of the prediction status and of the words already typed. At each keystroke it predicts suggestions, in the form of a list of word completions, by assuring accordance (gender, number, person, tense and mood) with the syntactic sentence context.

All the prediction functions are now encapsulated into a separate library, the *Predictive Module*, available also for integration with others applications. The *Predictive Module* provides core functionalities, such as the morpho-syntactic agreement and the lexicon coverage, efficiently accessing the *Linguistic Resources*, as the Language Model and very large lexical resources. We added new resources, namely POS *n*-grams and Tagged Word (TW) *n*-grams to our Language Model, improving the quality of morphological information available for the Prediction Engine. The tagged word *n*-gram model extends the typical word *n*-gram model [10] by adding POS information. For example a word bigram (w_{i-1}, w_i) is extended to a Tagged Word bigram (w_{i-1}, w_i, t_i), where t_i is the POS of w_i.

We introduce a new prediction algorithm for the Italian language based on Linear Combination [11]. The approach closest to ours is the one presented in [6], that is a linear combination algorithm combining POS trigrams and simple word bigrams. Our algorithm extends this model to cope with inflected languages, by combining POS *n*-gram models with tagged word *n*-gram models.

The Italian POS *n*-grams, approximated to $n = 2$ (bigrams) and $n = 3$ (trigrams) and tagged word *n*-grams, approximated to $n = 1$ (unigrams) and $n = 2$ (bigrams) have been trained from a large corpus created from newspapers, magazines, documents, commercial letters and emails.

The POS trigram model determines the most likely POS tags for the current word, given the two previous POS tags, if necessary backed up by POS bigrams. The TW bigram model establishes the most likely words given the immediately previous word. The probability S for the current word is the result of a weighed linear combination of the models:

$$S = \alpha \cdot \mathbb{P}(w_i \mid (w_{i-1}, t_i)) + \beta \cdot f(t_i, t_{i-1}, t_{i-2}) \quad (1)$$

where $\mathbb{P}(w_i \mid (w_{i-1},\ t_i))$ is the probability of the TW bigram (w_{i-1}, w_i, t_i), i.e. the probability of the next word being w_i, given that the previous word is w_{i-1} and the next word should have the t_i POS,

$$f(t, t', t'') = \begin{cases} \mathbb{P}(t \mid t', t'') & if\ \mathbb{P}(t \mid t', t'') > \vartheta \\ \mathbb{P}(t \mid t') & \text{otherwise} \end{cases} \quad (2)$$

and ϑ is the threshold empirically set. α and β are the coefficients of the linear combination and their sum must be 1 ($\alpha + \beta = 1$).

Donkey, the new FastType user interface (shown in Figure 2), is very simple and particularly easy to use. The system provides the user with a list of ranked suggestions. The user accept a word either by selecting the related function key (F1, F2, F3, and so on) or by using the pointing device (e.g. a traditional mouse or an eye tracker) to click the corresponding button. In this way the user can continue to write, looking for suggestions in the list and choosing the desired word that will be automatically inserted into the text.

Since there is a typical cognitive load associated to the interaction with word prediction systems due to the disability level or the limited language proficiency, Donkey can be adapted to the user needs. Donkey configuration utility provides a set of options that allow the user to personalize the word predictor functionalities, such as dimension, font, capitalization of the text in the suggestion list or its length.

Figure 2: *The User Interface.*

The length of the suggestion list influences the time and the effort required to search and select the right word. In consequence, the user can customize the number of suggestions in a range of 1-10. We have limited the number of suggestions presented by Donkey to 10, since a user can notice at a glance a word appearing in a smaller list, rather than in a larger list. Indeed, the larger the list, the higher the level of concentration required to read all the suggestions.

Donkey can be adapted even further in order to achieve a better interaction for blind or visually impaired users: for example Text-to-Speech options are available for reading words in the suggestion list or the selected word.

4. Evaluation

As described in [5], it is difficult to find appropriate metrics to measure prediction activities. In particular, a metric may be of more pertinence than another if there is an impairment in the user abilities. Thus we performed a general evaluation of the system, using different evaluation metrics.

Keystroke Saving (KS): being $c_1 \ldots c_n$ is an evaluation metric largely adopted in literature and provides a significative-for-all measure of efficacy. Keystroke Saving (KS) estimates the saved effort percentage and is calculated by comparing two kinds of measures: the total number of keystrokes needed to type the text (K_T) and the effective number of keystrokes using word prediction (K_E). Hence,

$$KS = \frac{K_T - K_E}{K_T} \cdot 100$$

There are two additional metrics we use to evaluate FastType prediction accuracy: Keystrokes Until Completion (KUC) and Word Type Saving (WTS).

Keystrokes until Completion (KUC): being $c_1 \ldots c_n$ the number of keystrokes for each of the n words before the desired suggestion appears in the prediction list,

$$KUC = \frac{(c_1 + c_2 + \ldots + c_n)}{n}$$

Word Type Saving (WTS): the percentage of time the user saves with FastType. Being T_n the time needed to write a text without FastType and T_a the time needed to write the same text with FastType,

$$WTS = \frac{(T_n - T_a)}{T_n} \times 100$$

To measure FastType performance improvements with the new linear combination algorithm we ran trials on the same test set presented in [12]. The test set was a subset of 40 texts disjoint from the training set. We developed a new test bench, performing different trials to experimentally determine the optimal value for α and β. The nutshell of the test bench is a 'simulated user' typing the test set and acting as a user that always selects the correct suggestion when predicted. We then measured the KS varying values for α and β. We ran trials increasing α by 0.1 from 0.1 to 0.9, empirically isolating the value of α producing the best KS.

Table 1: *Performance Measurement Results*

L	***KS***	***KUC***	***WTS***
5	46.79%	2.55	25.36%
10	51.16%	2.34	28.66%
20	55.13%	2.06	29.19%

A parameter that can greatly influence performance measurements is the length L of the prediction list, so we ran three trials on the test set with $L = 5$, $L = 10$ and $L = 20$. As we can see in Table 1, the increase in KS, WTS and KUC between $L = 5$ and $L = 10$ is way more relevant than the increase between $L = 10$ and $L = 20$.

The average KS is between 46.79% and 55.13%, marking a sensible improvement if compared with our previous results. Performances are significantly good for WTS, meaning that -at a standard speed and without any added cognitive load- saving in time is average around 29%. Particularly significant is also the KUC, meaning that the correct word is suggested after an average of 2.5 keystrokes for $L = 5$, 2.3 keystrokes for $L = 10$ and 2 keystrokes for $L = 20$.

Figure 3 presents a sample text: predicted keystrokes, blue marked, are 175 out of a total of 349 keystrokes, thus producing, in this case, a KS of about 50%.

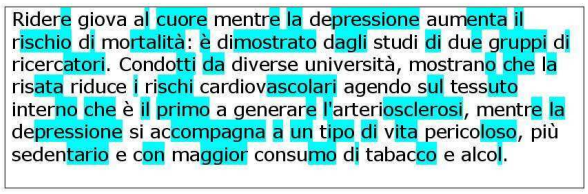

Figure 3: *Word Prediction results.*

Performances are comparable with existing works on non-inflected languages, as in [6] and [7], since with $L = 10$ FastType KS rises to 51%.

5. Conclusions

In this paper, we have presented the FastType system and its new human-computer interface, DonKey. DonKey allows the user to benefit from automatic word prediction in any desktop application.

We also described recent enhancements we introduced to the FastType system. By making use of POS tags we built a new Language Model and we refined the prediction algorithm.

We have evaluated FastType performance enhancements for an inflected language, i.e. Italian. According to our tests word prediction reaches a Keystroke Saving up to 51% for a standard prediction list of length 10. Keystroke Saving is now comparable to the one achieved by other systems for non-inflected languages, thus outclassing some typical word prediction limitations.

Our conclusions are consistent with state of the art literature, for example with [8], who claimed that a word prediction method without syntactic information are not applicable to inflected languages. We additionally enriched the Language Model with morpho-syntactic information and provided the prediction method with an on-the-fly Part-of-Speech word tagger and large lexicon dictionaries.

For future work we have plans for running field tests with disabled people, in order to improve DonKey usability in daily tasks as writing texts or emails and communicating. We have also plans for designing and developing a prototype version for PDAs and smartphones.

In conclusion, FastType has peculiarities and potential advantages since, using very large lexical resources and statistically based techniques, an effective word prediction can be performed in real domains.

We believe that the application of this technology is wide and we are working to bring the benefits of fast text typing to virtual keyboards and portable devices like smartphones and PDAs.

6. Acknowledgements

The FastType project is partially funded by the Fondazione Cassa di Risparmio di Pisa.

7. References

[1] A. Copestake, Augmentative and Alternative NLP Techniques for Augmentative and Alternative Communication, *Proceedings of the ACL Workshop on NLP for Communication Aids*, 37–42, 1997.

[2] C. Aliprandi, N. Carmignani, P. Mancarella and M. Rubino, A Word Predictor for Inflected Languages: System Design and User-Centric Inferface, *Proceedings of the 2^{nd} IASTED International Conference on Human-Computer Interaction*, 2007.

[3] I. MacKenzie, J. Chen and A. Oniszczak, Unipad: Single-Stroke Text Entry with Language-based Acceleration, *Proceedings of the Fourth Nordic Conference on Human-Computer Interaction*, 78–85, 2006.

[4] H. Trost, J. Matiasek and M. Baroni. The language component of the FASTY text prediction system. *Applied Artificial Intelligence*, 743-781, 2005.

[5] N. Garay-Vitoria and J. Abascal, Text Prediction Systems: A Survey, *Universal Access in the Information Society*, 4(3), 188–203, 2006.

[6] A. Fazly and G. Hirst, Testing the Efficacy of Part-of-Speech Information in Word Prediction, *Proceedings of the 10^{th} Conference of the EACL*, 2003.

[7] S. Palazuelos-Cagigas, J. Martín-Sánchez, L. Hierrezuelo Sabatela and J. Macías Guarasa, Design and Evaluation of a Versatile Architecture for a Multilingual Word Prediction System, Proceedings 10^{th} International Conference on Computers Helping People with Special Needs, 894–901, 2006.

[8] K. Tanaka-Ishii, Word-based Predictive Text Entry Using Adaptive Language Models, *Natural Language Engineering*, 13(1), 51–64, 2007.

[9] N. Garay-Vitoria and J. Abascal, Word Prediction for Inflected Languages. Application to Basque Language, *Proceedings of the ACL Workshop on NLP for Communication Aids*, 29–36, 1997.

[10] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, 1999.

[11] N. Deha, FastType: Predizione di Parola basata su Modelli Statistici in un Ambiente di Scrittura Assistita, Master's Thesis in Computer Science, Department of Computer Science, University of Pisa, 2007.

[12] C. Aliprandi, N. Carmignani and P. Mancarella, An Inflected-Sensitive Letter and Word Prediction System, *Proceedings of the International Conference on Interactive Computer Aided Learning*, 2006.

[13] J. Arnott, A. Newell and N. Alm, Prediction and Conversational Momentum in an Augmentative Communication System, *Communications of the ACM*, 35(5), 46–57, 1992.

[14] C. Aliprandi, N. Carmignani, N. Deha, P. Mancarella and M. Rubino, FastType, a Word Predictor for Inflected Languages: Syntactic Prediction Features and User-Centric Interface, *Proceedings of the 9th European Conference for the Advancement of Assistive Technology*, 378–382, 2007.