Posgram Driven Word Prediction

Carmelo Spiccia, Agnese Augello and Giovanni Pilato
Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), Italian National Research Council (CNR),
viale delle scienze, edificio 11, Palermo, Italy

Keywords: Word Prediction, Posgram, Part of Speech Prediction, Two Steps Prediction, Missing Word, Sentence Completion.

Abstract: Several word prediction algorithms have been described in literature for automatic sentence completion from a finite candidate words set. However, at the best of our knowledge, very little or no work has been done on reducing the cardinality of this set. To address this issue, we use posgrams to predict the part of speech of the missing word first. Candidate words are then restricted to the ones fulfilling the predicted part of speech. We show how this additional step can improve the processing speed and the accuracy of word predictors. Experimental results are provided for the Italian language.

1 INTRODUCTION

Predicting the missing word of an incomplete sentence through algorithms is a challenging task which has applications in Text Autocompletion, Speech Recognition and Optical Text Recognition. Both automatic and semi-automatic methods have been described in literature. A semi-automatic word prediction software was Profet (Carlberger et al, 1997). Being developed in 1987, it employed unigrams and bigrams. More recent approaches include neural networks (Mnih and Teh, 2012) (Mikolov et al, 2013), syntactic dependency trees (Gubbins and Vlachos, 2013) and Latent Semantic Analysis (LSA) (Bellegarda, 1998) (Zweig and Burges, 2011) (Spiccia et al, 2015).

In 2011 a training dataset and a questionnaire have been developed by Microsoft Research for its Sentence Completion Challenge (Zweig and Burges, 2011). Each question is composed by an English sentence having a missing word and by five candidate words as possible answers. This questionnaire simplifies the task in several ways. First of all, the five possible answers to a given question always have the same part of speech (e.g. adjective). Secondly, some parts of speech are never present in the answers set, stopwords in particular: conjunctions, prepositions, determinants and pronouns. Thirdly, in a real application the entire dictionary should be considered, not just five words. Therefore, real word applications involve processing a larger and more heterogeneous set of candidate words. To handle the general task better, we propose an innovative methodology for predicting a missing word of a sentence. We focused our study on the Italian language, even though the proposed approach is in principle general. The methodology consists in two steps. In the first step the number of candidate words is reduced. In particular, a novel algorithm based on posgrams has been developed for predicting the part of speech of the missing word. Candidate words can therefore be reduced to the ones fulfilling the predicted part of speech. In the second step a word predictor is applied on this reduced words set. This can be accomplished by using any of the word prediction algorithms described in literature. The following sections describe the proposed methodology in more detail, which is also illustrated in fig. 1.

Section 2 demonstrates why the two steps prediction is advantageous: formulae for the estimation of the success probability of the word prediction and for the estimation of the execution time reduction are derived.

Section 3 quantifies the advantages for the Italian language: a tagged corpus is parsed and statistics about each part of speech are collected; the a priori probability of each tag is estimated; the formulae derived in section 2 are then used to estimate the gains in terms of accuracy and execution time.

Section 4 describes how the part of speech prediction step can be accomplished: a novel algorithm based on posgrams is proposed.

Spiccia, C., Augello, A. and Pilato, G..
Posgram Driven Word Prediction.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 589-596
ISBN: 978-989-758-158-8

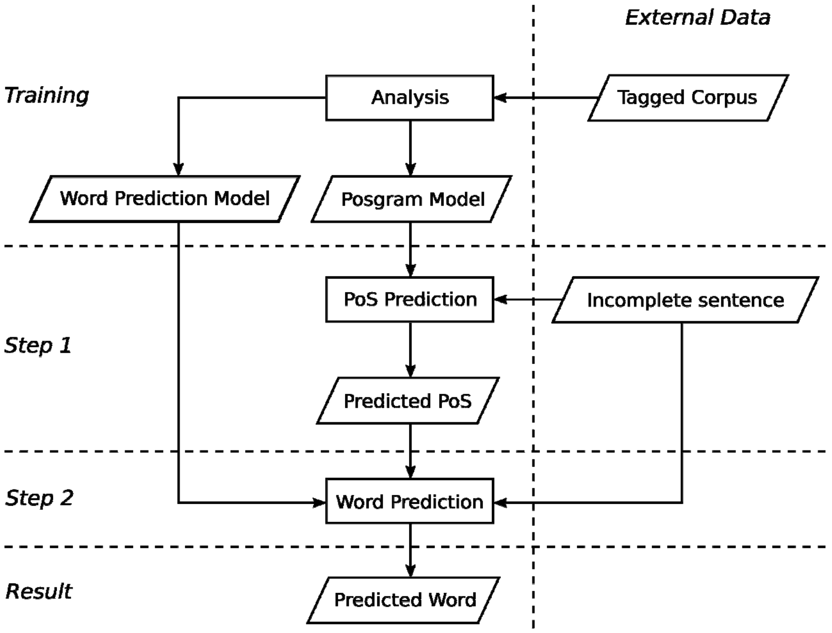

Figure 1: The two steps word prediction methodology.

Section 5 describes how to use the information achieved in the first step to predict the missing word: a training procedure is described to assert which is the best action to take for a given predicted part of speech.

Section 6 shows the final results: the accuracy of the part of speech prediction algorithm is compared to some baseline methods; the accuracy of the two steps word prediction method is compared to the single step method on a novel questionnaire of incomplete Italian sentences.

2 TWO STEPS PREDICTION OF A MISSING WORD

Given a sentence having a missing word, predicting the part of speech of the word before predicting the word itself may be convenient. To quantify this advantage, a question is created on the basis of that incomplete sentence, by adding *n* candidate words as possible answers. Let n_c be the number of candidate words having the part of speech *c*. Let us suppose that for any part of speech the number of words in the dictionary having that part of speech is far greater than *n*. When building the answer set, the effect of choosing a word on the probability that the next word will have the same part of speech can be therefore ignored. We will show that this hypothesis is conservative. If the part of speech of the missing word is *c*, the probability that the answers set contains *k* words having the same part of speech is:

$$P(n_c = k \mid c) \cong \binom{n-1}{k-1} P(c)^{k-1} \cdot (1 - P(c))^{n-k} \tag{1}$$

Suppose *c* is known when answering the question. This gives an advantage to the word predictor. Let *S* be the success at predicting the word. The probability of the event, by choosing a random word among the answers having the part of speech *c*, is:

$$P(S \mid c) = \sum_{k=1}^{n} \frac{P(n_c = k \mid c)}{k} \tag{2}$$

The success probability, regardless of *c*, will be:

$$P(S) = \sum_{c \in C} P(c) P(S \mid c) \tag{3}$$

where *C* is the set of the parts of speech. If the previous hypothesis is false, e.g. if the number of words of the dictionary having a specific part of speech is very small or if *n* is huge, the above formula will give a lower bound estimate of the success probability. This happens because the components *P(n*$_c$*=k|c)* are increasingly overestimated as *k* grows and they are weighted by the inverse of *k*.

Knowing the part of speech in advance gives two advantages. First of all, accuracy is improved since:

$$P(S) \geq \frac{1}{k} \tag{4}$$

The second advantage is the required execution time: by reducing the cardinality of the answers set, less candidate words must be evaluated by the prediction algorithm. In the next section we analyze these advantages in detail for the prediction of the missing word of an Italian incomplete sentence.

3 TWO STEPS PREDICTION FOR THE ITALIAN LANGUAGE

In order to apply the two steps prediction model to the Italian language, we consider the WaCky Italian Wikipedia Corpus (Baroni et al, 2009), freely available in the CoNLL format. It uses two tagsets conforming to the EAGLES standard (Medialab, 2009) (Stubbs, 2007): a coarse-grained one (14 tags) and a fine-grained tagset (69 tags). The first tagset is shown in Table 1. It assigns a letter to each part of speech.

Table 1: The coarse-grained tagset.

Tag	Part of speech	Tag	Part of speech
A	Adjective	N	Number
B	Adverb	P	Pronoun
C	Conjunction	R	Article
D	Determinant	S	Noun
E	Preposition	T	Predeterminant
F	Punctuation	V	Verb
I	Interjection	X	Other

For testing the model, we set to five the number *n* of candidate words per question. Table 2 shows, for each part of speech *c*, from left to right: the tag; the a priori probability *P(c)*, computed by analyzing the corpus; the probability $P(n_c=k|c)$ of obtaining *k* answers having the same part of speech of the missing word, for $k = 1 \dots 5$, conditioned to *c* being that part of speech; the probability of success *P(S|c)* at predicting the missing word by choosing among the answers having part of speech *c*, conditioned to *c* being the part of speech of the missing word; the probability of *c* being the part of speech of the missing word and to succeed, at the same time, at predicting that word.

The sum of the values of the last column is 0.7102. It represents the probability *P(S)* of success at predicting the missing word by using the two steps methodology. This means that by solving the problem of predicting the part of speech of a missing word, the problem of predicting the word among five possible answers is solved with at least 71% of accuracy. To provide a comparison, the current state of the art single step algorithm achieves an accuracy of 58.9% at predicting the missing word among five possible choices (Zweig and Burges, 2011).

Table 2: Probabilities related to the construction and usage of a five-answers question.

c	*P(c)*	$P(n_c=1\|c)$	$P(n_c=2\|c)$	$P(n_c=3\|c)$	$P(n_c=4\|c)$	$P(n_c=5\|c)$	*P(S|c)*	*P(c)P(S|c)*
S	0.2803	0.2683	0.4180	0.2442	0.0634	0.0062	0.5000	0.1401
E	0.1712	0.4719	0.3898	0.1207	0.0166	0.0009	0.6726	0.1151
F	0.1380	0.5521	0.3536	0.0849	0.0091	0.0004	0.7320	0.1010
V	0.1030	0.6474	0.2974	0.0512	0.0039	0.0001	0.7974	0.0821
A	0.0837	0.7051	0.2575	0.0353	0.0021	0.0000	0.8345	0.0698
R	0.0799	0.7166	0.2490	0.0325	0.0019	0.0000	0.8418	0.0673
B	0.0399	08498	0.1412	0.0088	0.0002	0.0000	0.9205	0.0367
C	0.0381	0.8560	0.1357	0.0081	0.0002	0.0000	0.9239	0.0352
P	0.0305	0.8834	0.1112	0.0053	0.0001	0.0000	0.9391	0.0287
N	0.0245	0.9056	0.0910	0.0034	0.0001	0.0000	0.9511	0.0233
D	0.0093	0.9631	0.0364	0.0005	0.0000	0.0000	0.9813	0.0092
T	0.0013	0.9947	0.0053	0.0000	0.0000	0.0000	0.9974	0.0013
X	0.0002	0.9990	0.0010	0.0000	0.0000	0.0000	0.9995	0.0002
I	0.0000	0.9998	0.0002	0.0000	0.0000	0.0000	0.9999	0.0000

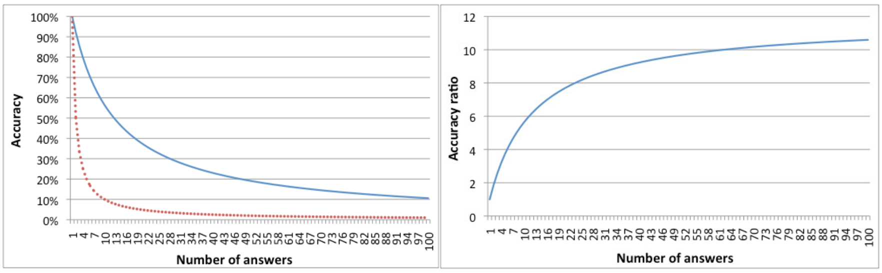

Figure 2: On the left: accuracy of prediction strategies as the size of the answers set grows; single step (dotted red) and two steps (blue) random choice word predictors are compared. On the right: accuracy ratio between the two steps and the single step prediction strategies.

Table 3: Parts of speech a priori probabilities and distribution.

C	P(c)	Words	Words %	P(c) · (Words %)
S	0.2803	44528	42.74%	11.98%
E	0.1712	32575	31.27%	5.35%
F	0.1380	25242	24.23%	3.34%
V	0.1030	1889	1.81%	0.19%
A	0.0837	751	0.72%	0.06%
R	0.0799	128	0.12%	0.01%
B	0.0399	116	0.11%	0.00%
C	0.0381	97	0.09%	0.00%
P	0.0305	93	0.09%	0.00%
N	0.0245	79	0.08%	0.00%
D	0.0093	18	0.02%	0.00%
T	0.0013	11	0.01%	0.00%
X	0.0002	7	0.01%	0.00%
I	0.0000	5	0.00%	0.00%

Fig. 2 shows the obtainable advantage in terms of accuracy at different answers set size *n* for the Italian language. The graph on the left compares the two steps model with the single step model: both models use random choice for word prediction, but the first model performs a part of speech prediction step to reduce the number of answers. Accuracy levels are reported. The graph on the right shows the ratio between the accuracy of the first model and of the second model. The two steps model always outperforms the single step model, with up to 10 times or greater accuracy.

Analyzing the first 200,000 words of the corpus, the number of unique words is 104,174, while the unique word-tag pairs are 105,539. Therefore, for a very limited number of words, up to 1.31%, more than one part of speech may apply. Table 3 shows, for each part of speech: the tag; the a priori probability; the number of unique words in the corpus; the percentage of unique words in the corpus; the product of the a priori probability with the percentage of unique words. The sum of the values of the last column is 20.95%. It represents the average percentage of the answers set to be processed when the set is a casual sample of the whole dictionary. This can lead to a speedup of about 5x (e.g. for LSA five times less scalar products must be computed).

4 PART OF SPEECH PREDICTION

In order to predict the part of speech of the missing word, the first 200,000 words of the corpus have been parsed with a moving window: the frequency of every sequence of one to five parts of speech has

Table 4a: Most common posgrams by length and their frequencies on the first 200,000 words of the WaCky Italian Wikipedia Corpus. The length spans from one to three.

1-posgram	Freq.	2-posgram	Freq.	3-posgram	Freq.
S	28039	ES	10072	SES	5201
E	16283	SF	7336	ESE	2657
F	13600	SE	7312	ESF	2587
V	10692	RS	6278	RSE	2260
A	8574	SA	4142	ESA	1678
R	8169	SS	3001	SAF	1663
B	4033	VE	2944	VRS	1606
C	4021	AF	2538	VES	1600
P	3175	SV	2385	ERS	1341
N	2200	AS	2277	SSF	1317

Table 4b: Most common posgrams by length and their frequencies on the first 200,000 words of the WaCky Italian Wikipedia Corpus. The length spans from four to five.

4-posgram	Freq.	5-posgram	Freq.
ESES	1835	SESES	844
RSES	1639	ESESF	529
SESF	1546	VRSES	457
SESE	1191	ESESE	453
SESA	850	RSESF	421
ESAF	733	SESAF	379
VRSE	713	RSESE	372
SAES	641	ERSES	366
ASES	536	VESES	336
SESS	519	RSESA	288

Table 5: Posgram windows, up to the length of five, and their centrality.

Window	Centrality	Window	Centrality	Window	Centrality
X	1	XX_	1	_XXXX	1
_X	1	_XXX	1	X_XXX	2
X_	1	X_XX	2	XX_XX	3
_XX	1	XX_X	2	XXX_X	2
X_X	2	XXX_	1	XXXX_	1

been saved to a lookup table. We will refer to these sequences as "posgrams" (Stubbs, 2007) (Lindquist, 2009). The most frequent ones are reported in Tables 4a-b.

Given a sentence having a missing word, the part of speech can be predicted by using the posgrams lookup table. The window of five words centered on the missing word is considered: each of the known words is replaced by the corresponding most common part of speech; the predicted part of speech of the missing word is the one which maximizes the frequency of the posgram. In order to improve the prediction accuracy in case of infrequent or absent posgrams in the corpus, an ad-hoc smoothing algorithm has been developed. First of all we define the "centrality" of a window with respect of the part of speech to be predicted as the number of parts of speech, plus one, between the missing element and the nearest extremity of the window. Table 5 shows the centrality of each window up to size five. The part of speech to be predicted is represented by an underscore; the known parts of speech are represented by an "X".

The pseudo-code on fig. 3 illustrates the smoothing algorithm. It takes in input three parameters: the maximum size *p* of a posgram; the extended window composed by the concatenation of the *p* – 1 tags on the left of the missing word, an

```
Function predictPos(maxPosgramSize, window, weights)
   bestScore = 0
   mostProbablePos = "S"
   w = 0
   For size = maxPosgramSize To 1
      For Each pos In tagset
         subwindows = findSubwindows(window, size)
         For Each subwindow In subwindows
            posgram = Replace "_" In window With pos
            scores[pos] += frequency(posgram)
                         * centrality(subwindow)
                         * weights[w]
            If scores[pos] >= bestScore Then
               bestScore = scores[pos]
               mostProabablePos = pos
            End If
         End For Each
      If bestScore > 0 Or w > 0 Then
         w = w + 1
         If w = length(weights) Then Break
      End if
   End For
   Return mostProbablePos
End Function
```

Figure 3: Pseudocode of the smoothing algorihtm employed for part of speech prediction.

underscore and the $p - 1$ tags on the right of the missing word; a vector α of weights. Subwindows of progressively lower size are processed: the score of each part of speech is incremented for each posgram found in the lookup table. The increment is the product of: the frequency of the posgram; the centrality of the subwindow; the weight α_i, where i is the difference between the size of the first posgram matched and the size of the current subwindow.

5 WORD PREDICTION

Given the predicted part of speech, this information can be used to improve the accuracy of the word prediction. First of all, it's convenient to assert the best action to take for each possible part of speech of the tagset. Therefore, the training phase is split into two steps. In the first step, the part of speech predictor and the word predictor are trained. In the second step, a questionnaire is automatically created from the corpus: from each sentence a question is built, by removing a random word; the other candidate words are chosen randomly from the nearby sentences; these words and the removed word constitute the answers set for the question. For each question the part of speech is predicted and the word predictor is invoked on the full answers set. Afterwards, the word predictor is invoked again on the restricted answers set composed by the words having the predicted part of speech. Results statistics are collected and aggregated by the predicted part of speech. After this second step of training, the achieved statistics provide information on which action to take. In particular they tell whether to restrict the answers set to the predicted part of speech, i.e. if the prediction of that part of speech is sufficiently reliable to actually improve word prediction.

6 RESULTS

The proposed part of speech prediction method employed on the first step is not directly comparable with general Part of Speech Tagging (POST) algorithms. In fact those algorithms are concerned with finding the most probable sequence of parts of speech for a complete sentence. While several approaches has been described in literature for handling unknown words, i.e. words not present in the training dataset, no studies have been done, at the best of our knowledge, on handling missing words. POST algorithms address a different task since they assume that no words are missing in the middle of the sentence. For unknown words, they generally take advantage of the word morphology, e.g. prefixes or suffixes, for predicting the part of speech. This cannot be done for missing words. Therefore, table 6 compares the proposed part of speech prediction method with two very baseline methods: random choice and choice of the most probable part of speech, i.e. noun ("S"). The best results are obtained with $p = 5$, $\alpha_0 = 0.5$, $\alpha_1 = 16.7$, $\alpha_2 = 0.2$, achieving an accuracy of 43.2%.

Table 6: Accuracy of various part of speech prediction methods for missing words.

Method	Accuracy
Posgrams	43.2%
Always noun	28.0%
Random choice	7.1%

State of the art algorithms for word prediction reported in literature have been tested with the Microsoft Sentence Completion Challenge dataset. This is currently the only complete training-test dataset specifically developed for measuring automatic sentence completion algorithms. However, because of its limits exposed in section 1 (uniformity of the part of speech in the answer set of any given question and unrepresented parts of speech among missing words), this dataset could not be employed. Therefore, in order to test the full word prediction methodology a new questionnaire has been built. Its format is the same of the one used for the Microsoft Sentence Completion Challenge: each question is composed by a sentence having a missing word and by five candidate words as answers. However our questionnaire is more general, since they address the aforementioned limits. First of all, each word of the same answers set may belong to a different part of speech. Secondly, the missing word can belong to any part of speech, including: conjunctions, prepositions, determinants and pronouns. The questionnaire has been built by selecting 368 random Italian sentences from the Paisà (Lyding et al, 2014) dataset. From each sentence a question is built, by removing a random word; the other candidate words are chosen randomly from the nearby sentences; these words and the removed word constitute the answers set for the question. We employed three different word prediction methods for the second step: ngrams, Latent Semantic Analysis (LSA) (Spiccia et al, 2015) and random choice. Table 7 shows the results in term of accuracy.

Table 7: Accuracy of various word prediction methods on the Italian questionnaire, with (2 steps) and without (1 step) employing the proposed part of speech prediction algorithm.

Method	Accuracy
ngrams (2 steps)	51.1%
ngrams (1 step)	50.3%
LSA (2 steps)	30.7%
Random choice (2 steps)	29.3%
LSA (1 step)	25.5%
Random choice (1 step)	20.4%

Each two steps method always provides better results than its single step counterpart. Since any part of speech, except punctuation, is admissible in a question answers set, most stopwords such as conjunctions, prepositions and determiners had to be included during the training phase. This has undermined the quality of the semantic spaces employed by LSA, leading to lower results than those reported in literature for English. Furthermore, the Italian language is more agglutinative than English: for example, the word “accettandoglielo” stands for “accettando esso da lui”, which means “accepting it from him”; one verb, two pronouns and a preposition are agglutinated into a single word. This hinders the performance of methods, like LSA, that attempts to find a single fixed-length encoding for such words: in fact, some information will be inherently lost, unless an ad-hoc preprocessing step is taken to split them. Since these kinds of words are very frequent in Italian, the obtained results are not directly comparable with those reported for the Microsoft Completion Challenge. Even though the problem negatively affects the two steps methodology too, we purportedly have not added the preprocessing step: this has allowed us to assess a lower bound for the prediction accuracy achievable by the methodology in the worst-case scenario. Results show that word prediction methods with lower accuracy exhibit greater improvements (+8.9% for Random choice) than methods with higher accuracy (+0.8% for ngrams). In general, the greater the accuracy of the word prediction method, the greater the part of speech prediction accuracy step is required to be advantageous. While this is not surprising, it should be noted that even a 50.3% accuracy word prediction algorithm (i.e. ngrams) can be improved by a 43.2% accuracy part of speech predictor: in fact, depending on the predicted part of speech the accuracy of the first step may be greater and therefore still advantageous; when this is not the case the part of speech prediction will be automatically discarded, as described in section 5.

7 CONCLUSION

In this work we presented a word prediction methodology based on posgrams. The proposed approach differs from other algorithms for introducing an additional preparatory step aimed at predicting the part of speech of the missing word. The number of candidate words can therefore be reduced accordingly. This has lead to an absolute accuracy improvement of up to 8.9% as shown by the experimental results. The methodology has been designed to automatically assess the reliability of the first step prediction: this allows to obtain small improvements even when the average accuracy of the part of speech predictor is relatively low. Future work will focus on adding preprocessing, improving the part of speech prediction step and exploiting other word prediction algorithms for the second step.

REFERENCES

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E., 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. In *Language resources and evaluation*, Vol. 43, no. 3, 209-226.

Bellegarda, J.R., 1998. A multispan language modeling framework for large vocabulary speech recognition. In *Speech and Audio Processing, IEEE Transactions on*, Vol. 6, no. 5, 456-467.

Calzolari, N., McNaught, J., Zampolli, A., 1996. EAGLES Final Report: EAGLES Editors' Introduction. EAG-EB-EI. Pisa, Italy.

Carlberger, A., Carlberger, J., Magnuson, T., Hunnicutt, S., Palazuelos-Cagigas, S.E., Navarro, S.A., 1997. Profet, A New Generation of Word Prediction: An Evaluation Study. In *Proceedings, ACL Workshop on Natural language processing for communication aids*, 23-28.

Ferraresi, A., Zanchetta, E., Baroni M., Bernardini S., 2010, Semantically and Syntactically Annotated Italian Wikipedia. WaCky Corpora. University of Bologna. http://wacky.sslmit.unibo.it/doku.php?id=corpora (Accessed on: 1st of July 2015).

Gubbins, J., Vlachos, A., 2013. Dependency Language Models for Sentence Completion. In *EMNLP*. 1405-1410.

Lindquist, H., 2009. *Corpus Linguistics and the Description of English*. Edinburg University Press, 102-103.

Lyding, V., Stemle, E., Borghetti, C., Brunello, M., Castagnoli, S., Dell'Orletta, F., Dittmann, H., Lenci, A., Pirrelli, V., 2014. The PAISA Corpus of Italian Web Texts. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9), 14th Conference of the European Chapter of the Association for Computational Linguistics*, 36-43.

Medialab, 2009, Tanl POS Tagset, University of Pisa. http://medialab.di.unipi.it/wiki/Tanl_POS_Tagset (Accessed on: 1st of July 2015).

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint, arXiv:1301.3781.

Mnih, A., Teh, Y.W., 2012. A fast and simple algorithm for training neural probabilistic language models. arXiv preprint, arXiv:1206.6426.

Spiccia, C., Augello, A., Pilato, G., Vassallo, G.: A word prediction methodology for automatic sentence completion. In *Semantic Computing (ICSC), 2015 IEEE International Conference on*, 240-243.

Stubbs, M., 2007. An example of frequent English phraseology: distributions, structures and functions. In *Language and Computers*, Vol. 62, no. 1, 89-105.

Zweig, G., Burges, C.J.C., 2011. The Microsoft Research Sentence Completion Challenge. *Microsoft Research Technical Report*. MSR-TR-2011-129.