The ABCP1 Language Model

Technical Report TR-ABCP1-02

Vitor M M C Pera

FEUP - Porto
June 2014

Abstract

The main goal of this report is to present in detail the language model (LM) of the ABCP1 speech recognizer. Here are reported several versions of the LM that were developed considering different options at the linguistic (lexical or syntactic) or implementation levels. This report also includes the explanation of experiments that were important on taking decisions that lead to some established solutions. When were recognized appealing research opportunities, sometimes these experiments were extended beyond the strictly necessary to build the LM. Briefly, the following achievements were attained. Both phonetically or syllabically based single-pronunciation lexicons, for a vocabulary not exceeding 20K words, were built. At the syntactic level, several probabilistic models were developed to implement the ABCP1 recognizer, or else to gain a better insight about this subject. Less common approaches were also experimented, for instance encoding into the grammars knowledge related to the lexical categories assigned to the words in the sentence, eventually considering gender and number concordance relations too. Interesting results were also achieved with methods that were experimented trying to reduce the impact of the lack of training data. One of these methods, which results seem more promising, starts training the LM with the available specific text corpora. Then, retrains a subset of the LM parameters identified as potentially less robust, using a much larger corpora, though possibly with quite different characteristics. For the best of the author's knowledge that approach presents conceptual originality and its natural implementation differs substantially of techniques used in Automatic Speech Recognition.

Contents

1 Introduction

It is well known that the performance of an automatic speech recognizer strongly depends on the efficiency of the respective Language Model (LM). In many speech recognition applications, language modeling becomes a difficult problem as soon as is intended to build an user friendly interface. Ultimately, this is a consequence of the great complexity of any natural language[1].

In the particular case of the speech interface for which the ABCP1 speech recognizer was built, some simplification was achieved imposing three main restrictions: 1) no spontaneous speech is allowed, though the user may pronounce the sentences continuously; 2) the vocabulary is limited to 20,000 words; 3) and is not accepted a large variation on the pronunciation of any word. Therefore, the ABCP1 system can be classified as a continuous speech, medium-to-large vocabulary and a speaker dependent like recognizer. Nevertheless, the following key points kept present all along the process of building the LM, that must comply with the established minimum performance level: 1) the need of building the LM as small as possible; 2) the need of having a LM fast enough to allow real-time operation; 3) and the interest in disposing of some adaptation method that could restrain the effects of the lack of data for training the LM. As an example, it was tried to replace higher order complete *ngrams* by smaller *ngrams* combined with less conventional linguistic knowledge that could be encoded efficiently. For instance, *bigrams* were combined with statistics modeling knowledge related to the lexical categories assigned to the words in the sentence, eventually considering gender and number concordance relations too. The empirical results were very interesting. Regarding to the point three, above, most of the work done addressed the problem of combining statistics estimated from a relatively small text corpus specific to this speech application, with those extracted from another corpus, which is much larger but presents very different linguistic contents. Contrarily to the standard approaches to this problem, the proposed method starts building a tuned LM using the specific corpus. And only then, an identified subset of the less well-trained parameters is retrained using also the larger corpus. The obtained preliminary results allow one to expect that this approach can effectively improve the LM, at least in some particular modeling contexts .

Reflecting the work done, contents of two different sorts compose this report: 1) the structure and the blocks implementation of any available version of the LM is here presented; 2) besides, the explanation of part of the experiments that were performed all along the work trying to find the answer to specific questions are reported here too. Such as it will be made clear in this report, a few of the proposed solutions are not intended to be

[1] When we study human language, we are approaching what some might call the "human essence," the distinctive qualities of mind that are, so far as we know, unique to man. *Noam Chomsky, Language and Mind.*

implemented in real-time in the ABCP1 recognizer, so a simulation must be carried out to assess its behavior.

The structure of this report is as follows. Section 2 presents a general description of the language models LM1 and LM2, corresponding respectively to the first and the second decoding stages of the ABCP1 recognizer. More detailed information, including related experiments, about the modules in these two models are given in the following sections. Section 3 is mostly dedicated to the word lexicons that were developed, based on phonemic, or else on syllabic, units. Section 4 addresses the LM at the syntactic level, describing the performed experiments, with the emphasis on the most relevant results, and also presents some selected versions of the grammars that were built for the LM1 and LM2 models. Suggestions for future work are presented in Section 5 and the conclusions are drawn in Section 6.

2 An overview of the LM1 and LM2 models

2.1 General aspects

The ABCP1 system supports a speech recognition application with the following features:

- is based on the European Portuguese [2] (EP) language;
- the recognition task presents broad linguistic topics;
- the vocabulary is limited to 20,000 words;
- the structure of most sentences corresponds to a reading-like continuous manner of speaking;
- and the pronunciation variation is small.

In Appendix B is presented a brief characterization of the main text corpus, named ABCP1_CP1, that was used to build this speech interface (more details can be found in [V. Pera (2011a)]).

The recognizer runs in real-time based on two decoding stages, or passes. The initial pass (pass-1), that can be viewed as performing a fast look-ahead operation, keeps uninterruptedly control of the decoding process during 2 seconds, approximately, in average. Then, the second pass (pass-2) gains the control during a shorter interval of time, eventually during a speech pause, and finds the best recognition hypothesis considering the search space that resulted of the pass-1. This cycle repeats as long as the recognizer operates.

[2] Even after the *Orthographic Accord* (2nd revision, 2008) was settled, some orthographic differences remain between the EP and the Brazilian one, for instance the pair *aritmética* and *arimética*, among other curious cases; around 0.5% of the general vocabulary of the language ($\sim$ 110K words) accepts double orthography.

These two passes require quite different restrictions, also at the language model level. The LM1 must contribute substantially to shrink the search space, obviously, but some trade-off was settled considering the gains that could result from using more linguistic knowledge and the available computational power. In brief, the acoustics play the main role in the established *fast look-ahead* approach. The fact that the recognizer is speaker dependent, which in general renders more reliable acoustic cues, naturally increases the importance of the acoustic (or visual, in this particular recognizer) restrictions. Accordingly, relatively costless linguistic restrictions were encoded into the LM1 model by means of: a phonemically based lexicon; and a *word-pair* grammar in combination with *unigram* statistics. Comparatively, the LM2 encodes deeper linguistic knowledge from the established recognition task. At the lexical level, two lexicons were built. One is based on phonemes and considers two-sided contextual information, so is a tri-phones lexicon. The other lexicon is based on *visyllables* (meaning "visual syllables"). The established set of *visyllables* should be minimum and still allow to represent properly any pair of syllables, considering the respective visual realizations. These two lexicons must be used simultaneously during the pass-2, when the acoustic and the visual feature streams are jointly decoded. At the syntactic level, the LM2 can use a bigram or a *trigram*. In some versions of the LM2, these grammars are combined with higher-order *ngrams* modeling the sequences of lexical categories, or *parts of speech* (POS), assigned to the words in the sentences.

2.2 The LM1 model

At the sub-word level, the LM1 is based on the 38 phonemes presented in the Table 1. This phonemic set is composed of 15 vowels, 3 glides and 20 consonants[3].

Each vocabulary word maps into an unique entry in the lexicon, according to the typical pronunciation with normal, or eventually slightly lower, speech rate. A few different words, such as for instance *á*, *há*, or *ah*, present identical pronunciation (/a/, in these cases). Two versions of this lexicon exist. One is a standard linear lexicon, where each word is transcribed separately. The other lexicon is supported by an hybrid structure: most of the words are transcribed based on a lexicon tree; and each one of the remaining words, that are selected based on recognition performance and operation speed criteria, is transcribed separately. In this report, the name of any version of this lexicon is LM1_LEX_v*n*. More detailed information is presented in the Section 3.1.

At the syntactic level, the LM1 consists of two main components, a *unigram* and a *word-pair* grammar. In the case of the words transcribed in

[3]Appendix A presents the correspondence between the IPA symbols and the symbols (similar to those in the SAMPA set) that are used in the developed code.

Symbol	Example
a	p<u>á</u>
ɐ	c<u>a</u>ma
ɐ̃	m<u>ã</u>o
ɛ	p<u>é</u>
ɛ̃	b<u>em</u>
e	d<u>e</u>do
ẽ	t<u>em</u>po
ɘ	d<u>e</u>dal
i	<u>i</u>da
ĩ	<u>in</u>do
ɔ	p<u>ó</u>
o	b<u>o</u>lo
õ	s<u>om</u>
u	t<u>u</u>do
ũ	m<u>un</u>do
j	pa<u>i</u>
w	ma<u>u</u>
w̃	mã<u>o</u>
b	<u>b</u>ota

Symbol	Example
p	<u>p</u>ó
d	<u>d</u>ata
t	ma<u>t</u>o
g	<u>g</u>ato
k	mo<u>ç</u>a
m	ca<u>m</u>a
n	<u>n</u>ome
ɲ	vi<u>nh</u>o
v	<u>v</u>ida
f	<u>f</u>ome
z	ca<u>s</u>o
s	tu<u>ss</u>a
ʒ	<u>j</u>antar
ʃ	me<u>x</u>e
ʎ	vel<u>h</u>o
l	mo<u>l</u>a
ɫ	me<u>l</u>
ɾ	faze<u>r</u>
R	co<u>rr</u>er

Table 1: The phonemes at the basis of the LM1.

the lexical tree, the *unigram* probabilities are factorized along the respective branches, following a standard approach. Naturally, in the case of the words associated to the linear lexicon, these probabilities are established at the entry points. In this report, the module with all these probabilities is referred as LM1_1G. More information on this grammar exists in the Section 4.2.1.

Regarding to the *word-pair*, it must be emphasized that during the pass-1 the decoding process is implemented assuming the *single one-best* hypothesis, strongly reinforcing the *bottleneck* effect associated to the lower nodes in the lexicon tree. Then, since the larger part (or even the entire vocabulary, if pretended) is supported by the lexicon tree, the *word-pair* restrictions cannot be fully profitable. Nevertheless, published results respecting to similar approaches show that even with that handicap the *word-pair* contributes effectively to reduce the recognition task perplexity and the respective *word error* rates. This grammar is named LM1_WP and is detailed in the subsection 4.1.

Table 2 summarizes some information in relation to the LM modules introduced in this Section, also linking them to the different versions of the LM1 that were built [4] (this information can be especially useful when

[4]It must be noticed that some implementation differences also exist, between the LM1-

Module	Brief description	LM1-v1	LM1-v2
LM1_LEX_*1*	linear lexicon (whole vocabulary)	•	
LM1_LEX_*2*	linear+tree lexicon		•
LM1_1G	*unigram*	•	•
LM1_WP	*word-pair*	•	•

Table 2: Information on the LM1 modules.

reporting experiments and results).

2.3 The LM2 model

When the pass-2 is running, the search space is established based on a *word-trellis*, containing all the candidate words. So, the main purpose of the LM2 is to assist, jointly with the audio-visual models, on the multi-modal decoding process running upon that *word-trellis*.

In a more simplistic approach, the pass-2 would dispense additional lexical knowledge, since each word candidate already has assigned the respective acoustic likelihood computed in the pass-1. The same lexicons used in the LM1 can be used here. In the pass-2 may be necessary to re-score the acoustic likelihoods at the words ends, due to the eventual (small) differences in the alignments, as a measure to mitigate the effects of the *single one-best* hypothesis assumed in the pass-1. Finally, the pass-2 also uses the visual features stream, which implies to know how to compose any word based on an effective set of sub-word *visual units*. According to these main principles, two main lexicons support the LM2 model: the LM2_LEX_PHN and the LM2_LEX_VSL.

The LM2_LEX_PHN is based on the same 38 phonemes used in the LM1_LEX (see Table 1).

The LM2_LEX_VSL is based on sub-word units that are defined at the syllabic level. In this report these units are named *visyllables*, or *visual syllables*, since they correspond to the visual realization of the respective syllables. For instance, /pa/$_v$ could denote the *visyllable* corresponding to the sequence of frames showing the mouth of the subject when utters the final syllable of the word *mapa* (*map*). One obvious advantage of this approach, when designing a speech recognizer, is that the number of *visyllables* is much smaller than the number of syllables. By the other side, it can happen that different words, or parts of words, present an identical *visyllabic* representation even when they are acoustically distinguishable. For instance, the words *mapa* and *papa* are very difficult to distinguish visually. Both words can be *visyllabically* transcribed as /pa-pa/$_v$ (in each word, the initial vowel is more open than the other one, but visually that

v1 and LM1-v2 versions, in the modules LM1_1G and LM1_WP.

difference vanishes). In Portuguese there are more than 4,000 syllables and the number of vysillables is just in the order of the hundreds. Comparing to what happens at the phonetic level, in Portuguese usually are considered between 30 and 40 phonemes and only 13 visemes. In spite of these consequences for the discriminability loss of the visual stream, it is important to have in mind that the visual cues are fundamentally supplementary to the acoustics. The following steps were performed to build the *visyllabic* transcription of each word in LM2_LEX_VSL: 1) get the phonetically based transcription (possibly discarding the contextual information); 2) apply a syllabification procedure in order to obtain the syllabic transcription; 3) finally, using a phoneme-to-viseme table, convert this segmented sequence of phonemes into the vysillabic transcription of the word. In the second step, above, the procedure was almost manual, but larger lexicons would obviously obligate to automatize the syllabification process. It is intended in a near future to improve the obtained *visyllabic* lexicon by means of known automatic techniques that use the available visual material associated to the text corpus (see Section 5).

At the syntactic level two different approaches were followed. One is simply based on *bigrams* or *trigrams*. Several versions were built, with different combinations of the smoothing method, the *cutoff* parameters, the text source or the vocabulary. The fact that the available corpus is relatively small caused some difficulties even in the case of the *bigrams*. Obviously, the *trigrams* were much more affected by the lack of data. Indeed, none of the *trigrams* could be developed following the standard procedures. No exclusive validation or test sets were established and the *trigrams* were trained simply using the same data that is used to perform the experiments. Nevertheless, it is important to emphasize that these *trigrams* are important to support comparative experiments with very low perplexity. In this report, these *bigrams* and *trigrams* are named LM2_2G_v*n* or LM2_3G_v*n*, respectively.

The other approach is based on the combination of two knowledge sources. One ot these sources is essentially the same that is used in the LM2_2G_v*n*, again with the already experimented robustness difficulties. The other knowledge is related to the statistical dependencies observed in the respective sequences of syntactic classes. A syntactic tagger was chosen [5] to assign the *parts of speech* (POS) tags to the preceding words. Given the small number of different classes, even when also is associated the gender and the number information, *trigrams* to "predict" these classes can be trained more robustly. Besides, this knowledge is relatively independent from the speech application, so, such as expected, much larger corpora can be advantageous even when their linguistic scope is substantially different. In this report, these grammars are referenced by the name LM2_HG_v*n* (HG stands for *hybrid grammar*).

[5] More information on this tool can be found in the Appendix D.

Module	Brief description	LM2-			
		-v1n	-v2n	-v3n	-v4n
LM2_LEX_PHN	phonetic lexicon	•	•	•	•
LM2_LEX_VSL	*visyllabic* lexicon		•	•	•
LM2_2G_vn	2-grams	•	•		
LM2_3G_vn	3-grams			•	
LM2_HG_vn	hybrid grammar				•

Table 3: Information on the LM2 modules.

Table 3 summarizes some information in relation to the LM modules indicated in this Section, also linking them to the different versions of the LM2 that were built (this information can be useful when reporting experiments and results).

3 The lexical level in the LM

3.1 The LM1_LEX lexicon

The LM1_LEX lexicon consists of phonemically based transcriptions of all the words in the vocabulary. At present, two versions exist, the LM1_LEX_v1 and the LM1_LEX_v2. Both versions were built based on the ABCP corpus (including the acoustic materials), which main features in this context are: 1) the vocabulary consists of 7,599 words in the EP language ; 2) the pronunciation variation is rather small and corresponds to a reading-like continuous manner of speaking, with a normal or slightly slow register (all the acoustic material was produced by an unique subject). More information concerning this corpus can be found in the Appendix B and a more detailed presentation exists in the respective Technical Report[V. Pera (2011a)].

According to the features above, the LM1_LEX_v1 and the LM1_LEX_v2 are single-pronunciation lexicons. Both are based on the 38 phonemes (15 vowels, 3 glides and 20 consonants) presented in the Table 1. Although it is intended that a later version of the recognizer will support out of vocabulary (OOV) words, none of the existing LM1_LEX versions contains any entry dedicated to OOV models.

The LM1_LEX_v1 is a standard linear lexicon, with each word being transcribed separately.

The LM1_LEX_v2 lexicon is supported by an hybrid structure. The larger part of the vocabulary is usually transcribed based on a lexicon tree. A standard linear lexicon is used for the remaining words, usually function words and also other frequent words.

The Appendix E contains the information needed to have access to the data file with the phonemic transcriptions that both LM1 lexicons use, and

also to the code that compiles that information into the lexicon tree, in the case of the LM1_LEX_v2.

3.2 The LM2_LEX_PHN lexicon

For the time being the LM2_LEX_PHN lexicon is exactly equal to the LM1_LEX. Some modifications, such as allowing multiple pronunciations for some of the most common words, are intended to implement in the future.

3.3 The LM2_LEX_VSL lexicon

The LM2_LEX_VSL lexicon consists of the vysillabic transcriptions of all the words in the vocabulary. No pronunciation variation is considered, according to the existing phonetic lexicons. For the English language several suggestions exist for viseme sets, but the scenario is very different in the case of the Portuguese. So, a brief study was carried out in order to establish a visemes set that could be appropriate when building the LM2_LEX_VSL lexicon. In the case of the consonant sounds, for instance, Table 4 presents information, in particular concerning the articulation place, related to their grouping into viseme classes.

Consonant(s)	Artic. place	Artic. manner
b p	bilabial	oral stop
d t	alveolar	
g k	velar	
m	bilabial	nasal stop
n	alveolar	
J	palatal	
v f	labiodental	fricative
z s	apical	
Z S	palatal	africate
L	palatal	lateral
l h	alveolar	
r R	alveolar	vibrant

Table 4: Articulation place and manner of the consonants (ABCP symbols set).

The procedure used for building the LM2_LEX_VSL requires the phonetic lexicon LM1_LEX_1 and the phoneme to viseme mapping that is shown in the Table 5, based on the ABCP phonemic-symbols set.

Although the properties of the syllables go beyond just the phonetic segmentation of speech, the baseline procedure used for building the *visyllabic* transcription of each word, w, in the vocabulary presents the following steps:

Phoneme(s)	Viseme
a 6 6∼	a
E e e∼ @ y	e
i i∼ j	i
O o o∼	o
u u∼ w w∼	u
b p m	p
d t n	d
g k	g
z s	z
l h	l
r R	r
J L	J
Z S	Z
v f	v

Table 5: The phone/viseme conversion table (ABCP symbols set).

1) get the phonetic transcription, P_w, from LM1_LEX_1; 2) segment P_w according to the established syllabification principles, obtaining S_w; 3) using the phoneme to viseme conversion table, convert S_w into V_w. Considering, for instance, the word *tudo*, the results in each step are: 1) /t u d u/; 2) /tu-du/; 3) /du-du/$_v$.

In relation to the step (2), some clarification of the syllabification method is presented next. For the existing LM2_LEX_VSL, the segmentation was made non-authomatically, following an intuitive criterion according to a few principles that are summarized next.

Following a generally accepted model, also valid in the EP language, the structure of the syllables is hierarchical, such as Figure 1 shows.

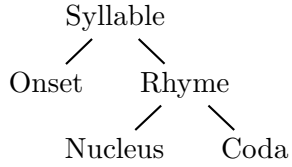

Figure 1: Syllables internal structure.

Every syllable has the *Rhyme* component. The *Rhyme* consists of the *Nucleus* and, possibly, the *Coda*. It can be that the *Rhyme* is preceded by the onset.

This structure complies with the general sonority principle: the sonority of the elements that compose a syllable increases from the beginning

up to the *Nucleus*, and then decreases until the end of the syllable. The sonority scale, by ascending order and in terms of broad classes, is: plosive consonants; fricatives; nasals; liquids; glides; and vowels.

Concerning to the onset position, it can be empty, e.g. in the monosyllabic word *ar* /a ɾ/ (*air*). Another possibility is to be simple, with almost any consonant, e.g. in the word *bar* /b a ɾ/. Or else, the onset position can be complex, with a sequence of consonants, e.g. in *traz* /t ɾ a ʃ/. In this last case, besides the sonority principle also must be satisfied, with just a few exceptions, the so called dissimilarity principle, stating that the distance between the sonority of contiguous consonants in the sequence must be maximum. Therefore, for instance, syllables with plosive followed by liquid in the onset, e.g. in the initial syllable of *prata* /pɾa-tɐ/ (*silver*), are much more frequent than syllables with plosive followed by nasal, such as in *gnose* /gnɔ-zə/ (*gnosis*). In the EP language, quite often occur three consonants in the onset, such as, for instance, in the initial syllable of the *estrada* (*road*), with the sillabically segmented phonetic transcription /ʃtɾa-dɐ/ (curiously, if the same word is uttered more slowly the transcription possibly becomes /eʃ-tɾa-dɐ/). These sequences can present up to six consonants, such as the initial syllable /dʃprʃti/, in the word *desprestigiar* (*depreciate*).

Relatively to the *Nucleus*, in the case of the EP language is always occupied by one or more vowels or nasals. An example of a simple *Nucleus* is the monosyllabic word *pé* /pɛ/ (*foot*). The complex *Nucleus* consists of a decreasing dithong (vowel+glide), eventually nasal, e.g. in the word *mo* /mɐ̃w̃/ (*hand*). Such as it was already exemplified, sometimes the syllabification is ambiguous, depending on the speech rate. For instance, in a normal register the word *guio* (*guidon*) presents two syllables, /gi-ɐ̃w̃/, with the *Nucleus* /ɐ̃w̃/ in the second syllable. But in a faster regist only one syllable exists, /gjɐ̃w̃/. Softening the rules, it can be considered that the nucleus is maintained and the glide /j/ joins /g/ in the onset position. When the vowels /i/ or /u/ succeed other vowel, eventually they do not become a glide, e.g. in the word *raíz* /Rɐ-Iʃ/.

Finally, in relation to the *Coda*, in the EP language often /ɬ/, /ɾ/ or /ʃ/ are the phonemes that occupy this position, e.g. in the words *mal* /maɬ/ (*wrong*), *mar* /maɾ/ (*sea*) and *mas* /mɐʃ/ (*but*). In the case of words ending with /e/, when this vowel is suppressed then the position of *Coda* is taken by the preceding consonant, e.g. in the word *bate* /bat/ (*beats*).

The syllabic annotation and segmentation information in the particular case of the ABCP corpus corresponds to a speech rate range between the slow and the normal registers. Relevant information and data concerning the ABCP, such as for instance the syllabic schemes (based on the Consonant or Vowel broad classes) or the *visyllables* frequencies, are presented in the respective Technical Report[V. Pera (2011a)].

In the Section 5 are suggested several approaches that possibly allow to improve the LM2_LEX_VSL lexicon. For instance, the problem of optimizing

the vysillables set and the transcriptions based on the existing visual material (eventually adapting known algorithms already experimented in the case of the acoustically based lexicons) should deserve particular attention.

The Appendix E contains the information needed to have access to the LM2_LEX_VSL data and the code used to generate this lexicon.

4 The syntactic level in the LM

4.1 The LM1_WP grammar

The LM1_WP is the *word-pair* grammar that was built for the APCP_CP1 corpus. All the 8 524 sentences of this corpus were used, such as usually. Besides the 7 599 vocabulary words, according to the defined lexicons, the sentence ends tags '<s>' and '</s>' are also included in this grammar.

The branching factor (not considering the sentence ends tags) is in the range $[1, 918]$, with 3 879 words (51% of the vocabulary) allowing only one different succeeding word. The normalized histogram of the branching factor is presented in Figure 2, confirming that most of the words, by far, have not many possible different successors. The word *e* is that one having more

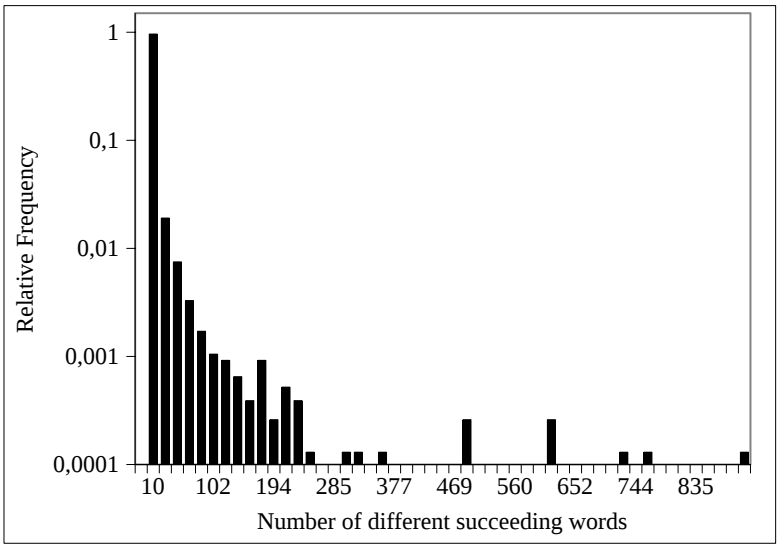

Figure 2: Normalized histogram (with log scale) of the word branching factor in the ABCP corpus.

different successors. Most of the other words with more successors are also function words, besides a few proper names. The number of different words that can start any sentence is 972, and 3 022 different words can be located at the sentences final position.

This grammar perplexity, estimated using all the sentences too, values 82. It must be emphasized that this value is a quite good estimate only in the case of using the LM1_LEX_1 (see LM1_v1 in Table 2). Otherwise, if using the hybrid lexicon LM1_LEX_2, this estimated value is just an optimistic lower bound of the *word-pair* perplexity, essentially due to the share of subword units in the nodes at the lower levels of the lexical tree.

The file *word-pair.dat* (UTF-8 encoding) uses a typical format, that can be very easily interpreted, to represent the LM1_WP grammar. The Appendix E contains information needed to have access to that data file and also to the programs used to encode that information in the hybrid lexicon.

4.2 The *ngrams*

4.2.1 The LM1_1G grammar

Both available versions of the LM1 use the LM1_1G *unigram*, that was built for the APCP_CP1 corpus, using all the available text (it was verified that only minor differences existed, such as expected, when using only the sentences in the ABCP_a data subset). In the case of the version LM1_v1, these probabilities just need to be available in a data file to be used directly when decoding. Otherwise, the version LM1_v2 is based on a lexicon tree, so in that case the *unigram* log-probabilities must be factorized and compiled into that data structure.

Approximately 48% of the words is the vocabulary (3 640 words) occur only once in the whole text. The word *que* is the most frequent, occurring 2 638 times, corresponding to approximately $3,3\%$ of all the words in the text. Figure 3 shows that the *unigram* log-probabilities distribution is approximately exponential. The perplexity of this grammar values 771, according to the estimate on the whole text. In the case of the LM1_v2, this value is just a reference.

The file *1gram.dat* (UTF-8 encoding) contains the LM1_1G probabilities. The Appendix E informs how to access that data file and also the programs used to compile the factorized *unigrams* into the hybrid lexicon.

4.2.2 The LM2_2G grammars

The LM2_2G grammars were built using the CMU-Cambridge Statistical Language Modeling Toolkit v2[6] (LMtk)[P. Clarkson (1997)]. These grammars were built for the ABCP corpus, which text material, that in this report is named ABCP, is splet into the ABCP_a and ABCP_b disjoint data subsets (see the Appendix B). To train each grammar, one of the following data sets was used: 1) the ABCP_a, that was established for that purpose; 2) or all the ABCP sentences. As a consequence of the lack of

[6] `http://mi.eng.cam.ac.uk/~prc14/toolkit.html`

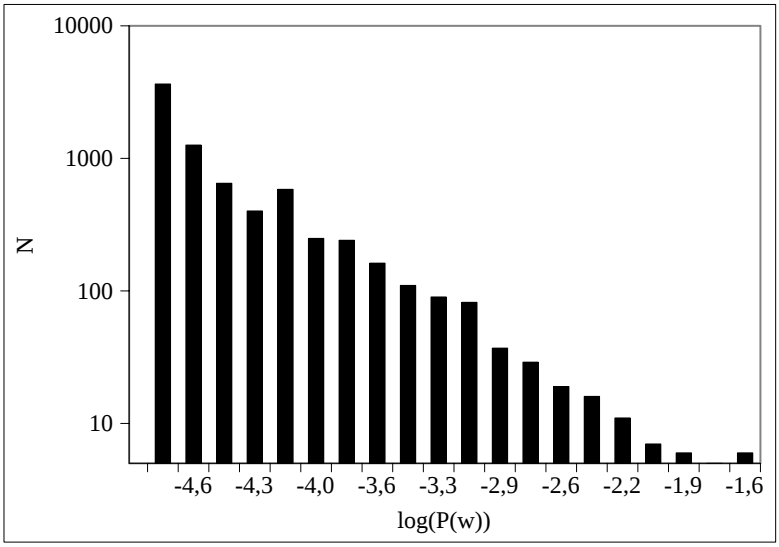

Figure 3: Histogram of the LM1_1G log-probabilities.

robustness, due to the quite limited training data, the initial option leads to relatively high perplexity (PP) values, when estimated on the ABCP_b data (see Table 6). The second option leads to more typical PP values on similar speech recognition tasks, establishing a reference (PP values in Table 7) that can be useful when experimenting the speech recognizer or when comparing with results obtained using the LM2_3G grammars. Concerning to the LM2_2G grammars that were trained using the ABCP_a subset, in the versions here reported the vocabulary is open according to the (LMtk) *-vocab-type=2* option, so that OOV is supported on testing although the vocabulary considered on training covers entirely the training sentences. In the versions trained with the whole data, a closed vocabulary model was built (*-vocab-type=0*). Here are reported the grammars that were built using the *Witten-Bell* or the *Linear* discount strategies, which in most of the performed experiments lead to PP estimates at the same level than the other two available options (*-good-turing* and *-absolute*). Different configurations of the *-cutoff* option were experimented, and results obtained with the values '0' or '1' are reported here. The begining of sentence symbol '< s >' was set in the *-context* option, allowing the respective forced back-off when estimating the PP. In Appendix E is given the information needed to have access to the *perl* script that builds these versions of the LM2_2G grammar.

Next are presented results, based on the ABCP_b testing data, that were obtained with different training configurations according to what was exposed before. In each sentence, the *unigram* probability was used for the initial word. The LMtk default value of any parameter was used, except if

explicited here on the contrary.

Table 6 shows the PP estimates when the models are trained with the ABCP_a data, leading to a 6.27% OOV rate. The results show that these

Smooth (option)	Perplexity
linear	316.27
witten-bell	308.41

Table 6: Perplexity estimates on the ABCP_b text, for different smoothing methods; train with the ABCP_a sentences (*vocab-type=2* option).

grammars are not able to decrease the recognition task difficulty as much as it could be expected, considering that the size of the vocabulary is not very large and the corpus linguistic scope can be considered quite typical, such as the sentences structure (the results in Table 7 help to clarify this aspect).

Often, when dealing with *ngrams*, another important aspect is the size of the models. Although nowadays, for implementations on standard PCs, the difficulties related with this subject generally only become relevant in the case of *trigrams* or higher order *ngrams*, it can be interesting to observe just a few results. The grammars which results are presented in Table 6 use approximately 130 MB (corresponding to approx. 33K *bigrams*). That space can be decreased in more than 77% (to $\simeq$ 30 MB) if the *bigrams* occurring only once are discarded, causing an increase of the estimated PP between 7% and 10%, depending of the smoothing method (from 316.27 to 337.43 or from 308.41 to 333.34 in the case of the *linear* or the *witten-bell* options, respectively). In terms of the number of *bigrams* hit in the ABCP_b data, as a consequence of setting *-cutoffs=1* the percentages change from 51.5% to 40.9%. In fact, most of the less frequent *bigrams* in the train data are not seen in the test data (as a reference, that percentage takes the value 89.6% for the respective closed vocabulary models built with all the ABCP sentences and using *exclusive back-off* option for the sentence begin).

Mainly with the goal of having some reference values to use with the LM2_3G grammars, *bigrams* were also built using all the sentences in the ABCP data set. Table 7 shows the PP estimates for different smoothing methods, when the models are trained that way. These results, combined

Smooth (option)	Perplexity
linear	72.70
witten-bell	65.40

Table 7: Perplexity estimates on ABCP_b, for different smoothing methods (train with ABCP).

with those presented in Table 6, reflect the fact that the ABCP_CP1 corpus is too small to support the robust train of a bigram (another experiment, with a different partition of the whole data into training and testing subsets, led to similar results). Concerning to this aspect, results obtained with the PUBLICO corpus, which total number of words in the selected sentences (see Appendix C) is 500 times (approximately) the total number of words in the ABCP data, reveal, such as expected, that the perplexity estimates are very close when the testing data is already known of the model (1.33% OOV rate, with the option *vocab-type=1*), or, on the contrary, was not seen yet; in the case of the Witten-Bell discount that difference is smaller than 0.2% and in the case of the other smoothing methods the difference keeps very small.

Comparing the PP values in Table 7 with the PP estimate respecting the LM1_WP grammar (also built using all ABCP sentences), that has the approximate value 82, somewhat surprisingly the performance indicators of the *word-pair* and the *bigrams* are quite close (in the case of the *good-turing* or the *absolute* discount options, not referred in this report, the *word-pair* PP is even slightly smaller, so that the effect of the discounts surpasses the advantage of using non-uniform distributions).

The Arpa-files corresponding to the six *bigrams* here referenced by the names LM2_2G_v1 to LM2_2G_v6 are already available. Appendix E shows how to have access to these files. Table 8 gives the information needed to link the different versions of the LM2_2G grammar to the respective training options and also recalls the respective estimated perlexity values.

Version	Train	Smooth (option)	Cutoffs	Perplexity
LM2_2G_v1	ABCP_a	*linear*	0	316.27
LM2_2G_v2	ABCP_a	*witten-bell*	0	308.41
LM2_2G_v3	ABCP_a	*linear*	1	337.43
LM2_2G_v4	ABCP_a	*witten-bell*	1	333.34
LM2_2G_v5	ABCP	*linear*	0	72.70
LM2_2G_v6	ABCP	*witten-bell*	0	65.40

Table 8: Information related to the already available LM2_2G grammars.

4.2.3 The LM2_3G grammars

Such as it was already pointed out, these grammars were built with the unique purpose of disposing of low perplexity values, necessary to perform some experiments implicating also other modules of the ABCP1 recognizer. The LM2_3G grammars were built using the LMtk package too. Given the obvious lack of data to train robustly these grammars, were only built versions trained with data containing all the sentences used on testing. Here

are reported the versions trained with the whole available sentences in ABCP These models contain approximately 70K *trigrams* when *cutoffs* are not applied. The vocabulary is closed (*-vocab-type=0*) and the beginning of sentence symbol '< s >' was established in the *-context* option, allowing the respective forced back-off when estimating the PP.

Table 9 shows the PP estimates on the ABCP_b data when the models are trained with different smoothing methods (using all the ABCP sentences on training and with the *default* configurations when not explicited otherwise). All the smoothing methods lead to low PP estimates such as expected

Smooth (option)	Perplexity
linear	26.95
witten − bell	10.71

Table 9: Perplexity estimates on ABCP_b, trainning with ABCP and different smoothing methods.

since the models already know the testing data, which is approximately 1/5 of the whole training data. It becomes clear that the perplexity corresponding to the *witten-bell* option is much lower than the other values. Possibly, in part this is associated to the tendency of the Witten-Bell smoothing to apply particularly aggressive discounts to the *ngrams* with higher values, what can bring some benefict when the training data is very scarce, such as happens here. Due to the *exclusive-backoff* option, approximately 80% of the 3grams in the model were hit on the test subset.

The grammars corresponding to linear or to Witten-Bell smooth are available. Table 10 links the versions names with the respective training options (also recalling the respective estimated perplexity values). Appendix E gives the indications needed to have access to the respective arpa-files.

Version	Train	Smooth (option)	Cutoffs	Perplexity
LM2_3G_v1	ABCP	*linear*	0; 0	26.95
LM2_3G_v2	ABCP	*witten-bell*	0; 0	10.71

Table 10: Information related to the already available LM2_3G grammars.

4.3 An hybrid grammar

4.3.1 Idea and formalism

Such as it was already pointed out, essentially the idea is to try to "compensate" some of the modelling losses due to using a lower order *ngram* (in

the case, use a bigram instead of a *trigram*) by means of imposing other linguistic restrictions.

This idea was implemented according to the following formalism. Begin supposing that the decoding process is in progress and let assume that the sequence of words in the sentence is statistically governed. Let the following discrete-valued random variables be stated. The variable w stands for the actual word hypothesis and w_{-k} denotes the k^{th} word preceding w (for instance, w_{-1} is the word immediately before w). The variables m, g and n denote, respectively, the lexical category (or *part of speech*), the gender class (possibly with the value "neuter") and the number inflection (possibly with the value "neuter") assigned to w. In the case of the word w_{-k}, these classes are represented by m_{-k}, g_{-k} and n_{-k}. The variable h denotes the sequence of words preceding w, from the begining of the sentence, and h_k is the sequence $\{w_{-k}, w_{-k+1}, \ldots, w_{-1}\}$ of the k words immediately preceding w. The sequence of *part-of-speech* tags corresponding to h is $\overline{m}_h$ and $\overline{m}_{h_k}$ denotes the sequence $\{m_{-k}, m_{-k+1}, \ldots, m_{-1}\}$. Given this, the conditional probability $P(w|h)$ can be expressed as follows:

$$\begin{aligned} P(w|h) &= \sum_m \sum_g \sum_n P(w, m, g, n|h) \\ &= \sum_m \sum_g \sum_n P(w|m, g, n, h) P(m, g, n|h) \end{aligned} \tag{1}$$

In general, the available data imposes some restrictions on the history lenght. In the particular case of the present LM and training corpus, the empirical results lead to the conclusion that when modeling the dependencies in the factor $P(w|m, g, n, h)$ the history h should consider only the word immediately preceding w. So, the following assumption is made:

$$P(w|m, g, n, h) \simeq P(w|m, g, n, w_{-1}) \tag{2}$$

It must be stressed that this approximation is quite crude, implying a substantial loss of discrimination ability, such as when a bigram is chosen in a standard approach. Concerning to the other factor in equation 1, the linguistic knowledge on the speech application allowed to introduce reasonable simplifications. Let begin factorizing that conditional probability as follows:

$$P(m, g, n|h) = P(m|h) P(g, n|m, h) \tag{3}$$

In relation to the factor $P(m|h)$, it can be assumed that the lexical classes assigned to the words in h convey most of the information existing in h about m. This assumption can be stated by the expression $I(m; h|\overline{m}_h) \simeq 0$. This means that the probability distribution governing m is almost independent of the value of h given the value of $\overline{m}_h$, so that $P(m|h) \simeq P(m|\overline{m}_h)$. Indeed, this approximation is as much acceptable as the conditional mutual information gets closer to zero. In the developed LM, $P(m|\overline{m}_h)$ can be trained

quite robustly with the history lenght up to the value 3. Assuming that this is enough to capture most of the information in $\overline{m}_h$, then:

$$P(m|h) \simeq P(m|\overline{m}_{h_k}), \quad \text{with } k \in \{1,2,3\} \tag{4}$$

The overall merit of this hybrid grammar depends in large measure of the accuracy of the $P(m|\overline{m}_{h_k})$ estimates. Obviously, if it could be possible to know somehow the exact value of m then the perplexity of this grammar would decrease substantially. Motivated by this, several experiments were performed addressing also the possibility of using a much larger corpus that would allow to train robustly a model for $P(m|\overline{m}_{h_k})$ with a relatively larger value for k. A brief description of that preliminary work is presented in the Subsection 4.3.2.

In relation to the other factor in equation 3, it was confirmed on the available data that is acceptable to consider g and n conditionally independent given m and h (it could be acceptable to consider g and n unconditional independent too), so

$$P(g,n|m,h) \simeq P(g|m,h)P(n|m,h) \tag{5}$$

Assuming that most of the information in h concerning the value of g given m is conveyed by the lexical class and the gender associated to the preceding word, then:

$$P(g|m,h) \simeq P(g|m,m_{-1},g_{-1}) \tag{6}$$

Since there was, from the beginning, some perception that the gender inflection of relatively distant words preceding w could carry important information about the value of g given m, a brief study was carried out in order to get a better insight on this topic. The obtained empirical results seem to confirm that the assumption supporting the model expressed by the equation 6 is quite reasonable, although it was also clear that it could be possible to optimize that model, at the cost of making it more complex and admitting that the training material would be enough to train it robustly. A brief description of that work is presented in Subsection 4.3.3.

Following an analogous reasoning to that expressed in the equation 6, it is made the assumption:

$$P(n|m,h) \simeq P(n|m,m_{-1},n_{-1}) \tag{7}$$

Therefore, the equation 5 is approximated such as follows:

$$P(g,n|m,h) \simeq P(g|m,m_{-1},g_{-1})P(n|m,m_{-1},n_{-1}) \tag{8}$$

Finally, gathering the results in the equations 2, 3, 4 and 8 into the equation 1, the following result is obtained:

$$P(w|h) \quad \simeq \quad \sum_m \left(P(m|\overline{m}_{h_k}) \sum_g \left(P(g|m,m_{-1},g_{-1}) \right. \right.$$

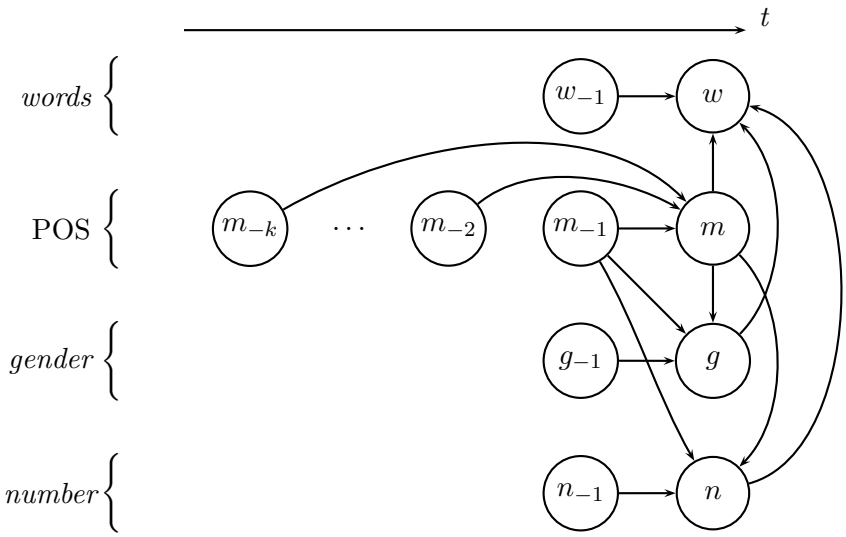

Figure 4: Structure of the developed probabilistic model.

$$\sum_{n} \Big(P(n|m, m_{-1}, n_{-1}) P(w|m, g, n, w_{-1}) \Big) \Big) \Big) \tag{9}$$

Figure 4 shows the structure of the developed probabilistic model, exposing the statistical dependencies, and also some of the independence assumptions that were made, among the established random variables. The causal reasoning that is clear in this representation is subjacent to the Bayesian Belief Networks (BBN) or, more generally, to the Graphical Models (GM). It is important to recall that not all the assumptions made were based on the observed statistical properties. In particular, the approximation expressed in the equation 2 was forced by the circumstance that the available text corpus is not large enough to allow training robustly a model with higher order dependencies. In the Section 5 are presented a few notes regarding to the possibility of using known structure learning (or model selection) methods in order to improve these models.

Concerning the implementation and parameterization of the probabilistic model, the task is relatively simple, since the variables are discrete-valued and all are observable. Based on the maximum likelihood (ML) criteria and on the relative frequencies, the local conditional probability distributions can be easily computed.

Empirical results were also obtained discarding the gender and number inflections associated to the words in the sentence (some of these results are

shown in Figure 7), in which case the equation 9 simplifies to:

$$P(w|h) \simeq \sum_{m} P(m|\overline{m}_{h_k}) P(w|m, w_{-1}) \tag{10}$$

4.3.2 Experiments with the morpho-syntactic model

This subsection reports some of the experiments that were carried out having in mind to improve the accuracy of the $P(m|\overline{m}_{h_k})$ estimates (in the sequence of the assumptions that lead to the equation 4), that are essential in this hybrid (or factored) grammar. These experiments followed two leading objectives. The first one, which is relatively simple, consists essentially on the assessment of these estimates, depending on the value of k (such as defined in the subsection 4.3.1) and based on the text corpus that was built for this specific application. That corpus is not large enough to allow robust training, and this fact is the main motivation for the other objective. Therefore, the experiments and respective results presented in the second part of this subsection deal with the problem of trying to combine, based on known principles, the original statistics with other extracted from a much larger corpus. Two main approaches were followed: one is based on the linear interpolation method; the other follows a back-off strategy, which is combined with additional restrictions designed to reduce the mismatch between both corpora.

The results that are going to be presented in this section are based on the ABCP_CP1 (see Appendix B) and the PUBLICO (see Appendix C) corpora. In concrete, are used the ABCP_a, ABCP_b, PBL_a and PBL_b subsets, containing 6 524 (66 003), 2 000 (21 244), 1 615 047 (32 007 253), and 461 422 (9 136 908) sentences (words), respectively. Ultimately, the efficiency of any LM in a speech recognizer is evaluated based on the recognition scores. That is not viable at the moment, and since in general exist a strong relation between these scores and the perplexity of the LM estimated on the testing data, in this work these estimates are used as the main indicators of the LM efficiency.

Figure 5 shows how the perplexity estimates obtained with two different models $P(m|\overline{m}_{h_k}) = P(m|m_{-k}, ..., m_{-1})$ evolve changing the value of $k \in \{1, \ldots, 4\}$. There are 17 different classes. In relation to the model trained using the ABCP_a dataset, are presented the curves corresponding to the perplexity estimated on the same data (ABCP_a curve) or else on the ABCP_b dataset (ABCP_b). It is clear from the curve ABCP_b that occurs overfitting when k exceeds the value 2. This phenomenon can be highlighted comparing with the behavior of the model that is trained with the PBL_a data (the curves PBL_a and PBL_b correspond to the perplexity estimates on these datasets). The curve PBL_b presents a monotonous behavior. The curves PBL_a and PBL_b almost coincide, apparently begining to diverge only when $k = 4$, also reflecting the relative training robustness, besides the

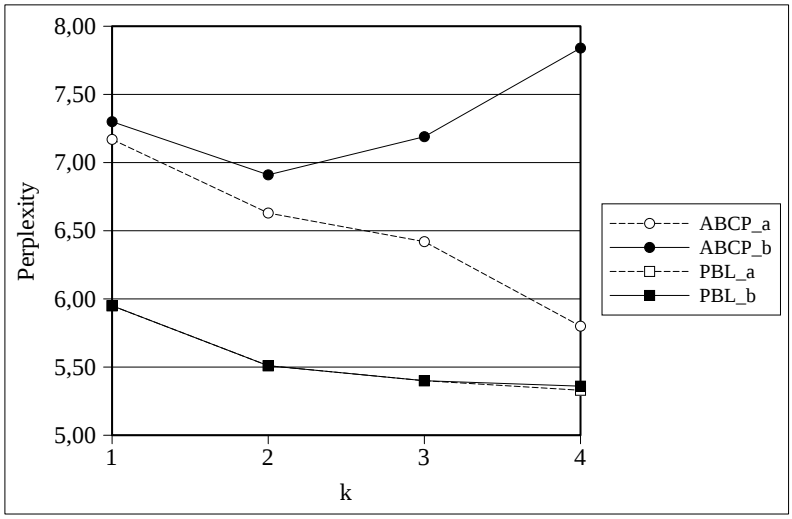

Figure 5: History lenght dependence of the perplexity associated to the models trained with the ABCP_a dataset (ABCP_a and ABCP_b curves) or with the PBL_a dataset (PBL_a and PBL_b curves).

non-existence of sensible mismatch between both datasets. One important conclusion emerges from these results: if the model is going to be trained using only the ABCP_a dataset, then must be fixed $k = 2$ (corresponding to a 3gram dependence). Indeed, this condition is applied in the LM2_HG_v2 grammar, that supported most of the experiments done, including those here reported, based on the ABCP_CP1 corpus in the scope of the *hybrid grammar* topic.

The results also suggest that, eventually, the model trained with the ABCP_a dataset can be improved using also information from the PUBLICO corpora. The perception of its potential is reinforced by the fact quite often emphasized that in general the *parts-of-speech* categorization is one of the linguistic analysis levels presenting lesser mismatch among text corpora.

One of the obvious possibilities to combine the statistics extracted from the two corpora is simply to interpolate linearly the respective models, according to the formula:

$$P(m|\overline{m}_{h_k}) = \lambda P_A(m|\overline{m}_{h_k}) + (1-\lambda)P_P(m|\overline{m}_{h_k}), \quad \lambda \in [0,1] \tag{11}$$

where $P_A(m|\overline{m}_{h_k})$ and $P_P(m|\overline{m}_{h_k})$ are the models trained, respectively, with the ABCP_a and the PBL_a datasets. Figure 6 presents the results that can be obtained with this approach. The three curves respect to the perplexity estimates (PP) on the ABCP_b dataset. The curve ABCP_b, respecting to the model $P_A(m|\overline{m}_{h_k})$, is already known from the Figure 5. The curve PBL_b, respecting to the model $P_P(m|\overline{m}_{h_k})$, behaves not completely surprisingly. Such as expected, the dependence on the value of k is monotonous in

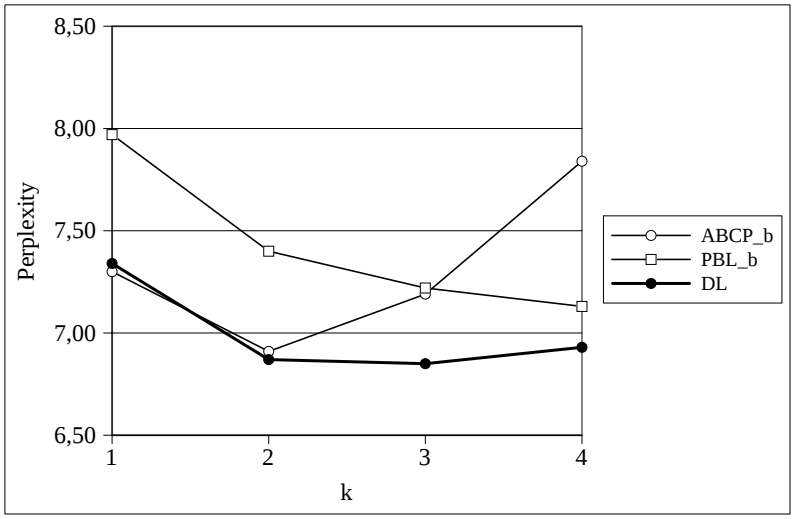

Figure 6: Perplexity estimations over the ABCP_b dataset using the models trained with the ABCP_a or PBL_a datasets (ABCP_b and PBL_b curves), or else with the interpolated model (DL curve).

this range of values. By the other side, the *parts of speech* linguistic level mismatch between the ABCP_a and the PBL_a datasets reveals to be substantial, such as suggested when $k = 1$. The curve DL (from *deleted interpolation*), corresponding to the interpolated model (equation 11), rises two interesting conclusions. One conclusion is that following this approach the statistics from $P_P(m|\overline{m}_{h_k})$ can improve clearly the behavior of $P_A(m|\overline{m}_{h_k})$ when k exceeds the value 2. In the particular case of $k = 3$, when the results from the ABCP_b curve only start degrading: the value of PP decreases from 7.19 to the 6.85 ($4.7\%_{rel}$ decrease). The other conclusion is that this approach is not able to reduce significantly the value of the perplexity when considering the whole k range. Indeed, the measured reduction (DL curve) form $PP = 6.87$, with $k = 2$, to $PP = 6.85$, with $k = 3$, is not significant. The fact that no significant gain can be obtained with this interpolation approach is due, in great part, to the substantial mismatch that exists between the ABCP_a and the PBL_a datasets at the linguistic level based on *parts of speech* (POS) tags (this suggests that one must be cautious when interpreting the common statement that "in general linguistic analysis based on POS categorization are relatively independent of the data").

Taking in consideration results that were obtained in analogous experiments, one cannot expect that the two main conclusions just exposed can change qualitatively if the interpolation uses different values of λ for each morpho-syntactic class, even being certain that some small improvement of the absolute PP estimates would occur. By the way, the values of λ used to

get the results in the Figure 6 are: 0.652 (if $k = 1$); 0.567 (if $k = 2$); 0.503 (if $k = 3$); and 0.429 (if $k = 4$). Not unexpectedly, the value of λ decreases (so, weighting more the larger dataset) when k becomes larger.

In the sequence of these results, the other approach that was followed is based on a back-off strategy, according to the equation:

$$P(m|\overline{m}_{h_k}) = \begin{cases} P_P(m|\overline{m}_{h_k}) & \text{if } \{m_{-k}, \ldots, m\}_P \text{ exists} \\ \epsilon(\overline{m}_{h_k}) P_A(m|\overline{m}_{h_{k-1}}) & \text{else} \end{cases} \tag{12}$$

Basically, the idea behind this approach is to try improving the results modeling higher range dependencies, based on more data, even running the risks inherent to the data mismatch. According to the proposed model, the estimates $P_P(m|\overline{m}_{h_k})$, based on the PBL_a dataset, are use if the respective sequences occur in that data. Otherwise, the lower order estimates $P_A(m|\overline{m}_{h_{k-1}})$, based on the ABCP_a dataset, are used. The $\epsilon(\overline{m}_{h_k})$ factor allows to get (entire) probabilities.

This model was experimented with $k = 3$, only, and the perplexity estimated on the ABCP_b dataset is 8.45. This result is clearly worse than the value of 7.19 (ABCP_b curve), corresponding to approximately $17.5\%_{rel}$ increase of the perplexity value.

Since the *parts of speech* linguistic level mismatch between the two corpora certainly is one of the leading factors for this bad result, it was tryied to get *normalized* upper branch conditional probabilities $\widehat{P}_P(m|\overline{m}_{h_k})$ imposing an additional condition:

$$\sum_{m_{-k}} \widehat{P}_P(m_{-k}, m_{-k+1}, \ldots, m) = P_A(m_{-k+1}, \ldots, m) \tag{13}$$

with the new estimates in the summation related with the old ones using the variable factor α as follows:

$$\widehat{P}_P(m_{-k}, \ldots, m) = P_P(m_{-k}, \ldots, m)\, \alpha(m_{-k+1}, \ldots, m) \tag{14}$$

Then, it can be easily deduced the adaptation formula:

$$\widehat{P}(m|\overline{m}_{h_k}) = \frac{P_P(m|\overline{m}_{h_k})\, \alpha(m_{-k+1}, \ldots, m)}{\sum_m \left(P_P(m|\overline{m}_{h_k})\, \alpha(m_{-k+1}, \ldots, m)\right)} \tag{15}$$

with α holding the quotient between the joint probabilities based on the ABCP_a or on the PBL_a datasets:

$$\alpha(m_{-k+1}, \ldots, m) = \frac{P_A(m_{-k+1}, \ldots, m)}{P_P(m_{-k+1}, \ldots, m)} \tag{16}$$

Replacing $P_P(m|\overline{m}_{h_k})$, in equation 12, by the adapted conditional probability $\widehat{P}_P(m|\overline{m}_{h_k})$, and considering $k = 3$, it is obtained the perplexity, estimated on the ABCP_b dataset, of 6.98. This result demonstrates the efficiency of the normalization method (equation 15), allowing to decrease the

initial probability value, 8.45, on approximately $17.4\%_{rel}$. Also compares favorably with the result from the model trained only with the ABCP_a dataset, for the same value of $k = 3$, decreasing the perplexity from 7.19, corresponding to $2.9\%_{rel}$ decrease. Finally, this result is worse than that obtained with the interpolation method ($PP = 6.85$, for $k = 3$), corresponding to $1.9\%_{rel}$ increase. It must be referred that using *in-house* developed code (see Appendix E), quite easily can be experimented a few variants based on the equation 12. A few suggestions concerning this topic are included in the Section 5.

4.3.3 Experiments with the gender inflection model

The main problem in consideration here is that of choosing good dependencies to reduce the perplexity associated to the gender inflection categorization (variable g) when using the model that computes the respective conditional probabilities (equation 9). It is well known, when choosing features in classification problems, that possibly "the m best features are not the best m features". If there is not enough aprioristic knowledge, it can be necessary to use some feature selection method. In this particular study was used an implementation of the method *mRMR - minimal redundancy maximal relevance*[H. Peng (2005), C. Ding (2005)] [7]. According to the authors, this implementation does not convolve with specific classifiers but one can expect that the selected features have good performance on various types of classifiers. In brief, following an incremental approach, each given feature is ranked based on the discriminative potential of that feature jointly with the features with higher rank, as a whole. Two evaluation functions are available, both based on the *minimum redundancy condition* ($\min\{W_I = 1/|S|^2 \sum_{i,j} I(i,j)\}$) and on the *maximum relevance condition* ($\max\{V_I = 1/|S| \sum_i I(g,j)\}$), where i and j are the candidate features, and g denotes the gender class. Those functions are: the *Mutual information difference criterion* (MID), $\max\{V_I - W_I\}$; and the *Mutual information quotient criterion* (MIQ), $\max\{V_I/W_I\}$. Table 11 presents part of the results generated by the program *mRMR*, based on the ABCP_a subset of the ABCP_CP1 corpus (see Appendix B). Two different cases are considered, such as the table shows: one considers that the gender variable can take the value *Feminin*, *Masculin*, or *Neuter*, that is, $g \in \{F, M, N\}$; in the other case, that is based on approximately half the data, are eliminated all the samples with *Neuter* gender, so $g \in \{F, M\}$. In each one of these cases, the features in the table are ranked, according to decreasing values of a score directly related to the mutual information $I(g; feature)$ measure. Some conclusions may be extracted from these results. Let start noting that when $g \in \{F, M, N\}$, the morpho-syntactic features m, m_{-1}, m_{-2} and m_{-3} (considering the history lenght 3), convey much more information concerning

[7] This program is available in the Web site `http://penglab.janelia.org/proj/mRMR/`

$g \in \{F, M, N\}$		$g \in \{F, M\}$	
feature	score	feature	score
m	1.003	g_{-1}	0.267
m_{-1}	0.151	g_{-2}	0.023
g_{-1}	0.137	m	0.009
m_{-2}	0.015	m_{-1}	0.006
g_{-2}	0.013	g_{-3}	0.003
m_{-3}	0.004	m_{-2}	0.002

Table 11: Features ranking based on $I(g; feature)$ related score (output of the program *mRMR*).

the value of g, comparatively with the other case. This is mainly a consequence of the fact that does not occurs the gender inflection in approximately half the lexical classes, particularly for the verbs, which are very frequent. Considering the case $g \in \{F, M, N\}$, that is the most important from the practical point of view, the obtained scores allow to conclude that m, m_{-1} and g_{-1}, by this order, are, clearly, the features carrying more discriminant information, when judged separately. Considering now the case $g \in \{F, M\}$, it can be verified that the gender inflection of the word preceding w, g_{-1}, is, *per se*, the most informative feature, presenting a score much higher than m or m_{-1}.

Table 12 presents other results generated by the program *mRMR*, based on the data already presented and considering $g \in \{F, M, N\}$. These results also consider the redundancy existing among the features, so they should contribute to clarify which feature set could be a good choice. A few inter-

MIQ		MID	
feature	score	feature	score
m	1.003	m	1.003
g_{-1}	0.630	g_{-3}	-0.003
m_{-3}	0.276	g_{-1}	0.021
m_{-1}	0.289	g_{-2}	-0.087
g_{-2}	0.100	m_{-3}	-0.276
m_{-2}	0.033	m_{-1}	-0.211
g_{-3}	0.012	m_{-2}	-0.404

Table 12: Features ranking based on the *minimal redundancy maximal relevance* principle, with $g \in \{F, M, N\}$ (output of the program *mRMR*).

esting conclusions can be drawn from Table 12. Such as expected, according to the results in Table 11, both MIQ and MID evaluation functions put m in

the first rank. Obviously, the following features deserve more attention since the value of m is supposed to be determined when computing $P(g|m,h)$ in equation 5. It is quite useful to know that, according to the MIQ evaluation, the feature g_{-1} is in rank 2, presenting the minimal redundancy and simultaneously the maximal relevance when considered jointly with m. Besides, if the MID evaluation function is used, still the feature g_{-1} is well classified, in rank 3. Considering for instance the MIQ based results and not regarding to m, it is clear the existence of three different score levels: g_{-1} at the higher level, then m_{-3} and m_{-1} at an intermediate level, and finally the lower ranked features g_{-2}, m_{-2} and g_{-3}. Still based on the MIQ results, this indicates that, for instance, the models $P(g|m,g_{-1},m_{-3})$ and $P(g|m,g_{-1},m_{-1})$ should be good approximations, at similar level, to the first factor in equation 5 and, on the contrary, $P(g|m,g_{-1},g_{-3})$, for instance, should be a worse choice. Curiously, making a similar analysis based on the MID evaluation function the conclusions concerning the use of the feature m_{-3} would be different. Indeed, some discrepancies between the results obtained with the MID or the MIQ functions are perfectly expected, since these functions use different operations to combine the *minimum redundancy* and the *maximum relevance* conditions and, besides, often the relative merits of the features are not very distinct (it must be noted that the *baseline* nature of these evaluation functions is emphasized by the authors). Nevertheless, it seems secure to conclude that the selection of g_{-1} as one of the features to use in the equation 5 is a good decision.

Table 13 presents some results that were generated by the program *mRMR*, based on the same data and considering again $g \in \{F, M, N\}$. The difference, comparatively to Table 12, is that these results were obtained categorizing jointly, for each word, the morpho-syntactic and the gender inflection (with exception of the word w), so, for instance, (m_{-1}, g_{-1}) denotes the random variable holding both properties for the word w_{-1}. Not regarding once again the feature m, the results respecting to the MIQ evaluation in Table 13 show that the m_{-1} and g_{-1} constitute a good subset of features to be selected. Considering the MID function, that pair of features seems to

MIQ		MID	
feature(s)	score	feature(s)	score
m	1.003	m	1.003
(m_{-1}, g_{-1})	0.485	(m_{-3}, g_{-3})	-0.018
(m_{-3}, g_{-3})	0.110	(m_{-1}, g_{-1})	-0.077
(m_{-2}, g_{-2})	0.059	(m_{-2}, g_{-2})	-0.536

Table 13: *Joint-features* ranking based on the *minimal redundancy maximal relevance* principle, with $g \in \{F, M, N\}$ (output of the program *mRMR*).

perform just little worse than (m_{-3},g_{-3}). In both cases, the pair (m_{-2},g_{-2}) is indicated as being the worst choice. Possibly, this result is affected by the *minimum redundancy condition*, that in this particular problem would favour some features spreading in the time domain.

Table 14 shows the results obtained by the *mRMR* program considering the *joint-features* such as those in Table 13, now resuming the experiment already reported in Table 11, when $g \in \{F, M\}$. These results strengthen

MIQ		MID	
feature(s)	score	feature(s)	score
(m_{-1}, g_{-1})	0.348	(m_{-1}, g_{-1})	0.348
(m_{-3}, g_{-3})	0.054	(m_{-3}, g_{-3})	-0.140
(m_{-2}, g_{-2})	0.042	m	-0.324
m	0.035	(m_{-2}, g_{-2})	$-.0538$

Table 14: Features rank based on the *minimal redundancy maximal relevance* principle, with $g \in \{F, M\}$ (output of the program *mRMR*).

the conviction that m_{-1} and g_{-1} constitute a good subset of features to be selected.

Indeed, the obtained results, including those here reported, reinforced by some practical aspects concerning the model implementation, contributed to the decision of selecting the m_{-1} and g_{-1} features to use jointly with m in the model expressed in equation 5. Briefly, this approach is simple to implement and seems quite efficient, although it becomes clear also that it would be possible to capture more information about the value of g given m using a more complex model.

4.3.4 Experiments with smoothing variations

Along this work, several experiments were carried out dedicated to the *ngrams* smoothing topic. Most of the work focused in *trigrams* $P(w|w_{-1}, m)$ (using variables definitions and nomenclature such as in the equation 10, for instance). Initially, the main motivation was to study possibilities of improving these models, considering aspects related to robustness and also accuracy, in the context of the distributions smoothing. Particular attention was paid to the integration of prior knowledge associated to the variable m, too. In the particular case of the Witten-Bell discount technique, was implemented and experimented a quite simple variation that allows to establish different discount levels. However, no systematization of these results was made that could deserve its presentation here. Besides, the main goal demands a deeper work, and there was no opportunity to do that effort yet. Anyhow, this introduction increased the confidence on the potential of this

subject and augmented the interest in continuing the work (in the Section 5 are made some references to this subject).

4.3.5 The LM2_HG grammars

Two implementations of the probabilistic models presented in Section 4.3.1 exist, namely the LM2_HG_v1 and the LM2_HG_v2. Both grammars are based on the equation 9, though, in this Section are also presented results corresponding to the model established according to the equation 10. The LM2_HG_v1 grammar was trained with the ABCP_CP1 corpus and the LM2_HG_v2 was trained with the PUBLICO corpus. According to the established model, besides the knowledge associated to the words sequences, from the data was also extracted the knowledge concerning the morpho-syntactic and the gender and number inflection (which categories are associated to each word in the vocabulary, possibly depending on the morpho-syntactic instantiation). To use these grammars, besides the data files containing the grammatical knowledge is also needed a POS tagger, that assigns the tags to words preceeding the current hypothesis during the model estimation. For the time being, the code allows to run *offline* a morpho-syntactic analyzer that is available in the Web (see the Appendix D to get more information on this tool). At least apparently, that tagger is based on a method that does not presents any essential obstacle to integrate it in an online version of the speech recognizer (see Section 5). Finally, is also available the code that integrates all the referred modules, allowing to simulate the grammar operation in the speech recognition process. Appendix E gives the indications needed to have access to these files.

In the case of the LM2_HG_v1 grammar, $n_m(n_g(n_n+1)+1)$ products and $n_m(n_g(n_n+1)+1)-3$ additions have to be computed to get each $P(w|h)$ estimate, where n_m, n_g and n_n are, respectively the number of different lexical classes (actually, 17), gender classes (3, including the *neuter*) and number inflection classes (also 3). In the Table 15 are presented the number of accesses (if no optimization method is implemented) to the hash-tables used to save the local conditional probabilities, and a lower bound to the respective sizes (that depends on using, or not, perfect hashing, or eventual approximations), where n_w is the vocabulary size and $n = n_m \,.\, n_g \,.\, n_n$. It is important to emphasize that the code can be optimized pruning the summations in equation 9, so that the terms corresponding to negligible conditional probabilities can be eliminated. It can be expected that the final results do not change significantly because of that approximation. The spatial requirements are dominated by the hash-table containing the estimates of $P(w|m,g,n,w_{-1})$, for moderate values of k.

In the case of the LM2_HG_v2 grammar, the time and space requirements are smaller than those respecting the LM2_HG_v1 grammar.

In the Appendix E are given the necessary indications to run the simu-

	$P(w\|\ldots)$	$P(m\|\ldots)$	$P(g\|\ldots)$	$P(n\|\ldots)$
Number of accesses	n	n_m	$n_m \,.\, n_g$	n
Size (upper bound)	$n \,.\, n_w$	$(n_m)^k$	$(n_m)^2 \,.\, n_g$	$(n_m)^2 \,.\, n_n$

Table 15: Hash-tables utilization in the LM2_HG_v1 grammar (k is the parameter in equation 9, typically an integer near 3) and the conditional probabilities refer to those in equation 9.

lations with the LM2_HG_v1 and the LM2_HG_v2 grammars.

4.3.6 Results obtained with the LM2_HG grammars

The results in this Section were obtained with the LM2_HG_v1 and the LM2_HG_v2 grammars, introduced in the previous Section. Table 16 presents two perplexity values, both estimated on the ABCP_b data. One refers to the baseline model that computes the conditional probabilities $P(w|w_{-1})$, therefore not using the POS categorization nor the gender and number inflection information (see Section 4.2.2). The other value was obtained with the LM2_HG_v1, considering $k = 2$ in the equation 9 (so, corresponding to a *trigram* to "predict" the variable m). The parameters of both grammars were trained using the ABCP_a data. These results, corresponding to

Model	Perplexity	(decrease)
$P(w\|w_{-1})$	308.41	(*baseline*)
LM2_HG_v1	280.57	($9.03\%_{rel}$)

Table 16: Baseline perplexity (PP) and PP based on the LM2_HG_v1 grammar (equation 9, with $k = 2$), both estimated in the ABCP_b testset.

a PP decrease in $9.03\%_{rel}$, are conform with the strong expectations that LM2_HG_v1 is more precise than the baseline grammar. Is also important to recall (Section 4.2.2) that the baseline grammar is not robust, so does not generalizes well. And it can be expected that these difficulties become worse in the case of the somewhat larger LM2_HG_v1 grammar. Therefore, the following experiments were obtained with models trained using a much larger dataset. Indeed, the LM2_HG_v2 grammar is based on the PUBLICO corpus. Is used the PBL_a subset, containing 1 615 047 (32 007 253) sentences (words), to train the models. And is used the PBL_c subset, with 230 720 (4 574 198) sentences (words), for testing[8], so that all the perplexity values here presented refer to this task. Changing to a much larger

[8] In substitution of the PBL_b subset, with approximately double size, initially used for testing, because of being very time consuming.

corpus brought, unfortunately, the disadvantage of having to account for a much higher computation effort to run the experiments. Because of this, it was decided to approximate the equation 9 in the implementation, reducing substantially the time required to run any experiment. This approximation allowed to eliminate two inner cycles in the code, in consequence of assigning to each word a category corresponding to m, g and n, jointly. So, mapping injectively (m, g, n) into a random variable c, is obtained the equation:

$$P(w|h) \simeq \sum_{c} P(c|\overline{c}_{h_k}) P(w|c, w_{-1}) \tag{17}$$

where $\overline{c}_{h_k}$ stands for $\{c_{-k}, c_{-k+1}, \ldots, c_{-1}\}$, following the established nomenclature. Obviously, the structure of this probabilistic model is more connected than that corresponding to the equation 9, since several assumptions introduced in the subsection 4.3.1 are not considered now.

The results in the Table 17 and in the Figure 7 refer to this implementation. To serve as reference, is presented in the top of the Table 17 the perplexity estimated with a standard *bigram*, $P(w|w_{-1})$, so not considering the lexical (m), gender (g) and number (n) categorizations. The vocabulary is open (according to the option *vocabulary-type=1* of the CMU/Cambridge-LM toolkit), considering only the subset from the 50K most frequent words of the corpus PUBLICO that also belong to the ABCP_CP1 corpus.

Configuration	Perplexity	(decrease)
$P(w\|w_{-1})$	144.92	(*baseline*)
HG w/ $k = 1$	129.08	($10.9\%_{rel}$)
HG w/ $k = 2$	120.13	($17.1\%_{rel}$)
HG w/ $k = 3$	126.82	($12.4\%_{rel}$)

Table 17: Perplexity estimates based on the LM2_HG_v2 grammar, for different values of k, in the equation 17.

Table 17 shows that the best result occurs with $k = 2$, corresponding to a substantial decrease of the perplexity ($17.1\%_{rel}$). If $k > 2$, then it seems that the model $P(c|\overline{c}_{h_k})$ becomes overfitting.

Results from two other tests, still based on the approximated implementation, are going to be presented next. One respects to the model expressed in the equation 10, not considering neither the gender nor the number inflections, relatively to the tests above (curve "m" in the Figure 7). The other test refers to the model expressed in the equation 9, but in this case not considering only the number inflection (curve "$m + g$" in the Figure 7). Figure 7 shows these results and also the baseline result and the other results in the Table 17 (curve "$m + g + n$").

In conclusion, the empirical results confirm clearly that is very advantageous, in terms of modeling precision improvement, to add morpho-syntactic

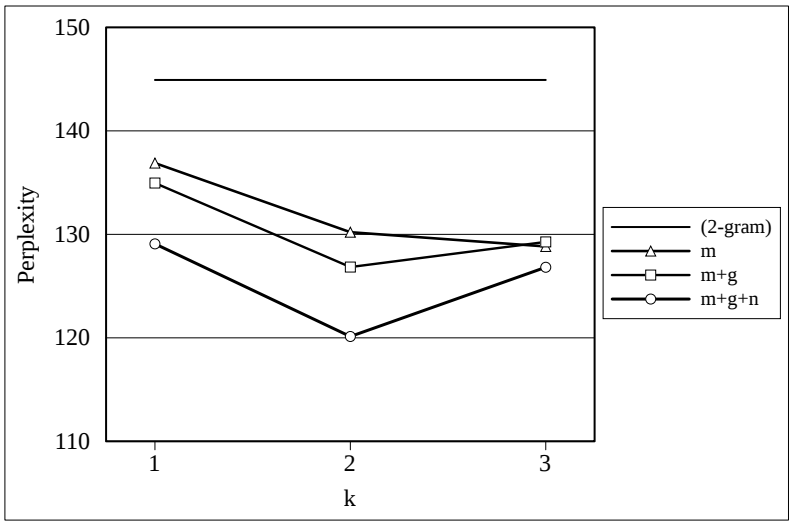

Figure 7: Perplexity estimates based on the LM2_HG_v2 grammar, for different values of k, in the equation 17, and different dependencies of the gender and number inflection categories.

and gender and number inflection knowledge to the standard LM approach.

4.4 A grammar adaptation mechanism

4.4.1 Introduction

Essentially, the problem here addressed consists of combining efficiently the knowledge extracted from multiple text corpora in order to build a good LM for a given application. It is intended that, for any sequence of words w, the probability $P(w)$ is a good estimate of the underlying real distribution. Let consider that two text corpora exist, encoding some linguistic knowledge associated to the application and differing essentially in two important aspects: the "suitability" in relation to the application, and the size. One of the data sets, let call it A, is specific to the application but is relatively small (a common problem in the case of the *ngrams*). The other, that can be denoted by B, is much larger but is less tuned with the application. Often, this problem appears in the context of an application which LM needs to be updated regularly. The existing model, that is supposed had be trained with a B-like corpus, is then adapted using the A-like data. In other contexts this approach is still frequent, although different intuitions could be more appropriate. In some particular cases the problem can be better viewed as departing from a LM trained with an A-like corpus, and at some point becomes available B, that hopefully could be used to improve the LM (let call

it "$A \rightarrow A + B$ approach) [9].

One of the interesting aspects of this approach is based on the possibility of identifying, using prior knowledge and having access to important cues resulting from the operation of the LM (trained with the A-like data), a subset of the model parameters particularly less robust. If the "weak parameters" of the original LM are effectively determined, then in principle better strategies can be implemented to improve the LM by means of knowledge extracted from the B-like data (see Section 5).

Several techniques have been proposed to deal with the problem of LM adaptation, which can be classified[J. Bellegarda (2001)] as *model interpolation* - including *model merging*, *dynamic cache models* and *MAP adaptation* -, or *constraint specification* - including *exponential models*, *MDI adaptation* and *unigram constraints* -, or *meta-information extraction* - such as *mixture models* or *explicit topic models*-. The techniques used in the experiments that were carried out, which are briefly presented in the next subsection, could be assigned to *model interpolation/MAP adaptation*. Also according to the taxonomy above, planned extension of these experiments, in the scope of the $A \rightarrow A + B$ idea, could include concepts and techniques from "*meta-information extraction*".

4.4.2 Experiments

In this subsection are going to be presented preliminary results obtained having in mind the $A \rightarrow A + B$ approach introduced in the previous subsection. So, it is supposed that already exists some relevant prior knowledge or some cues about the weaknesses of the LM (see subsection 4.4.1) trained with the A-like data. Due to practical reasons - one of the most important is that, at the moment, is not easy to simulate the integration of the LM in appropriate recognition processes to gather information that could help identifying some real weaknesses - and also due to the need of getting a better insight on relevant topics - for instance, concerning the use of syntactic knowledge -, the experiments here reported use a simplistic approach to the initial problem of identifying the "weak parameters". This allows to implement and simulate quite easily part of the idea .

Concerning the text data that supported the experiments, the scenario is as follows. In terms of training material, the ABCP_CP1 (Appendix B) and the PUBLICO (Appendix C) corpora support the A and B data sets, respectively. And the ABCP_b sentences set is used as the main testset.

The sentences from ABCP_a (the A data set) are supposed to be specific of the speech application, but are not representative enough. The lack

[9] Eventually, these "principle" differences vanish if some combination methods and respective implementations are followed; that can be the case, for instance, if using some interpolation methods (a quite common solution) to combine the parameters associated to A or B.

of robustness of the bigram corresponding to the conditional probabilities $P(w|w_{-1})$ and trained with the ABCP_a data is clear in the results shown in Table 8, for instance comparing the PP estimates obtained with the versions 1 and 5 (or 2 and 6). In the case of the respective *trigram*, with $P(w|w_{-1}, c)$ such as defined in the equation 17, even considering that the number of morpho-syntactic classes is relatively small (in the order of the tens, typically), it can be expected that the model robustness decreases even further.

Hope exists that the sentences from PBL_a (the B data set) contain usefull knowledge concerning this application. The size of PBL_a is between two and three magnitude orders larger than the size of ABCP_a. Table 18 presents results obtained with two *bigrams* trained with the PBL_a data, using the Witten-Bell discount method. In one of the *bigrams*, the vocabulary has size 57 K, containing all the words that appear at least 10 times in the data. The other *bigram* has a vocabulary with 5.7 K words, corresponding to all the words that exist in 57 K size vocabulary and also exist in the ABCP data. Considering the smaller vocabulary (that leads to high OOV

Vocabulary	Testset	Perplexity (bits)	OOV (%)	2grms hit (%)
5.7K	PBL_a	136.89 (7.10)	29.47	97.20
5.7K	PBL_c	145.85 (7.19)	29.48	95.60
5.7K	ABCP_b	487.56 (8.93)	10.06	82.66
57K	PBL_a	235.09 (7.88)	1.33	94.87
57K	PBL_c	344.93 (8.43)	1.43	86.62
57K	ABCP_b	629.56 (9.30)	5.09	79.21

Table 18: Perplexity, OOV and 2grams-hit rates for different testsets, depending on the vocabulary considered to build a bigram trained with the PBL_a data (Witten-Bell method).

rates), it is obvious that the PBL_a allows quite robust training of the model parameters, corresponding to only 0.09 bits entropy increase when testing the bigram on PBL_c instead of PBL_a. And the entropy increases 1.83 bits when testing the bigram in the ABCP_b data, reflecting the substantial mismatch between the PBL_a and the ABCP_b data sets. Considering the other vocabulary (approx. 57K entries), now the number of parameters in the ngram is approximately one order of magnitude higher. Therefore, a less robust model can be expected. Indeed, the entropy increases 0.55 bits on the PBL_c data. The increase in that measure remains high in the case of the ABCP_b testset. In this context, the main conclusion is that a substantial linguistic mismatch exists between the ABCP and the PBL data sets (both ABCP_a and PBL_b consists of large parts of ABCP and PBL, respectively). Besides, the results also give some intuition on the ability of the

PBL_a data set to train robustly a bigram. These can be useful indications, since in the model $P(w|w_{-1}, c)$ one of the variables is the morpho-syntactic class, existing in a small domain (and carrying relatively strong restrictions).

The experiments reported next consider the problem of improving the estimates of $P(w|w_{-1}, c)$, where w, w_{-1} and c denote, respectively, the current word, the previous word, and the morpho-syntactic class plus the gender and number inflections associated to w (such as appears in the equation 17). Let start considering that the available model to compute $P(w|w_{-1}, c)$ was trained with the ABCP_a data set. Given the serious limitations of this data set, already emphasized, some frequency smoothing method is crucial. A *trigram*, combined with a back-off smoothing scheme, was ued:

$$P(w|w_{-1}, c) = \begin{cases} \alpha(w_{-1}, c, w) & \text{if } \exists(w_{-1}, c, w) \\ \gamma(w_{-1}, c)P(w|c) & \text{else.} \end{cases} \tag{18}$$

In relation to the *bigram* corresponding to $P(w|c)$, in the performed experiments was also implemented the respective back-off smoothing scheme. In both the *ngrams* was applied the Witten-Bell discount method. The γ factor, in equation 18, (such as the analogous factor in the bigram) guarantee that $P(w|w_{-1}, c)$ sums to unity, for any (w_{-1}, c).

Now, according to the $A \rightarrow A + B$ approach, arises the question of identifying parameters in this model that can be specially responsible for its lack of robustness. Such as it was referred in the begin of this section, in the scope of these experiments a simplistic approach is proposed. In the model represented in the equation 18, some estimates of $\alpha(w_{-1}, c, w)$ are potentially more incorrect. In particular, an overestimation tendency generally affects the α parameters computed when both (w_{-1}, c, w) and $(w_{-1}, c, *)$ events ($*$ stands for *any class*) occur just once or a very few times. With the purpose of keeping the approach very simple, the experiments here reported considered for the selection criterion just the condition $\#_{(w_{-1},c,w)} = 1$. This selection criterion leads to a very simple implementation and has the potential to be effective enough, mainly for the purpose of enlighten this approach. It is important to stress that although this selection method demands only observing the specific training data, later is intended to consider other information sources, or cues.

Now emerges the question of how to use the sentences from PBL_a (the B data set) to improve the selected parameters subset. Often, the adaptation problem is put in terms of combining somehow two existing models, one trained with the A-data and the other trained with the B-data. Another possibility, also quite common, is based on the idea of combining the knowledge associated to both models (each one possibly existing only virtually) at the frequency count level, rather than at the model level. This second approach, that in general implementations has been associated with the *MAP* training criterion (and has been referred as *MAP estimation*), was chosen in these experiments. Three main arguments can be aligned favoring

this choice: 1) the interesting efficiency that has been reported (comparing to common approaches, such as those based on the linear interpolation of models); 2) the suitability of this method to this concrete strategy of selective adaptation, in terms of the parameters set (that is, becomes relatively easy to adjust only a subset of the parameters set using the new data); 3) and finally (this is an argument that has been emphasized to justify the performance of this approach), the subjacent MAP criterion establishes a framework allowing a "more principled way of combining LM information" comparing, for instance, with the linear interpolation of models isolated trained under the ML criterion.

In terms of the implementation of this approach, the results that are going to be presented were obtained using a very simple formula to adjust the selected α parameters (in the upper branch of the equation 18). According to previous explanation, it was decided to adapt only the parameters corresponding to sequences (w_{-1}, c, w) occurring just once in ABCP_a (the A-data) and that also occur at least one time (using of some arbitrariety) in the PBL_a sentences (the B-data), that is, in the equation 18 is used the formula

$$\alpha(w_{-1}, c, w) = \begin{cases} \alpha_A(w_{-1}, c, w) & \text{if } N_A > 1 \text{ or } N_B = 0 \\ \alpha_{A+B}(w_{-1}, c, w) & \text{else.} \end{cases} \tag{19}$$

Each α_A quantity results from the respective relative frequency, based on the ABCP_a data, and the Witten-Bell discount formula effect. And each α_{A+B} value results from a linear combination of the respective relative frequencies based on both data sets (no optimization weighting was performed for the baseline results) and is also affected by the Witten-Bell discounts. Such as it can be observed in the code (in the Appendix E are given the indications to have access to those files), the counts corresponding to the B-data are submitted to the logarithm transform, in order to compress and smooth somewhat the data (recall that the B-data is between two and three magnitude orders larger than the A-data), before the linear combination step is performed.

Table 19 presents the perplexity estimates on the ABCP_b data, relative to the model in equation 17, considering two versions of the $P(w|w_{-1}, c)$: the *baseline* version, trained with the ABCP_a data; and the *adapted* version, corresponding to train with the α parameters adjusted according to the equation 19. Several tests were performed in order to get more confidence in that the PP decrease effectively results of the extraction and use of knowledge encoded in the PBL_a data. Replacing the PBL_a data by small constant values, or by small random numbers (from an uniform distribution) added to the constant, always was verified the increase of the PP indicator. Besides, changing along reasonable ranges several parameters (not visible in the formula expressed in the equation 19) that allow to weight differently

$P(w\|w_{-1}, c)$	Perplexity	(decrease)
non-adapted	251.50	(*baseline*)
adapted	247.80	($1.47\%_{rel}$)

Table 19: Perplexity estimates, on the ABCP_b data, of the LM corresponding to the equation 17, for the *baseline* version of $P(w|w_{-1}, c)$ and for the *adapted* version.

the ABCP_a and the PBL_a influence, or that impose different levels at the Witten-Bell discount, always led to some reduction in the PP.

The perplexity reduction is small, only $1.47\%_{rel.}$. Nevertheless, it is perfectly reasonable to expect that the gain could be substantially larger if some conditions could be verified. First of all, obviously if the PBL_a data set presented linguistic characteristics closer to those existing in the ABCP_a data, then it would result much larger gains. This data mismatch, already emphasized above, is surely one of the main causes of the poor gain of the adaptation process. Another factor is related to the crudeness of the method used on selecting the *weak parameters* in the initial model. Such as it was already noted, intentionally a simplistic approach was followed for the selection step. And the combination formula is quite crude too, existing several possibilities to try sensible improvements. Besides, no optimization was performed in that formula, for instance weighting differently the ABCP_a and the PBL_a influences.

Figure 8 gives some intuition on the demanding task that is posed when trying to robust the α parameters, corresponding to the train with the ABCP_a dataset, using, according to the proposed manner, the counts from the PBL_a dataset. The coordinates of each point in the graphic are given by the number of occurrences of the respective sequence (w_{-1}, c, w) in the ABCP_a data (abcissas) and in the PBL_a data (ordinates). According to equation 19, the data corresponding to the points in the leftmost "column" (with Count in ABCP_a = 1) is specially relevant (for each point, the respective value appears in the numerator of the formula that computes the relative frequencies), when compared with the remaining points (the corresponding values are eventually "summed" in the denominator and so have a *smoothed* influence). And this "column" shows (although not revealing all about the distribution) that the occurrences of the (w_{-1}, c, w) events, which numbers in the PBL_a data spread along more than four magnitude orders, are crunched into an unique occurrence in the case of the ABCP_a data. Essentially, it is hoped that the distribution of all PBL_a data, and in particular that corresponding to the "column" at left, is more close to the applications real underlying distribution than the "relatively uniform" distribution, solely based on ABCP_a.

Among other aspects revealed in the graphic, is very curious (though

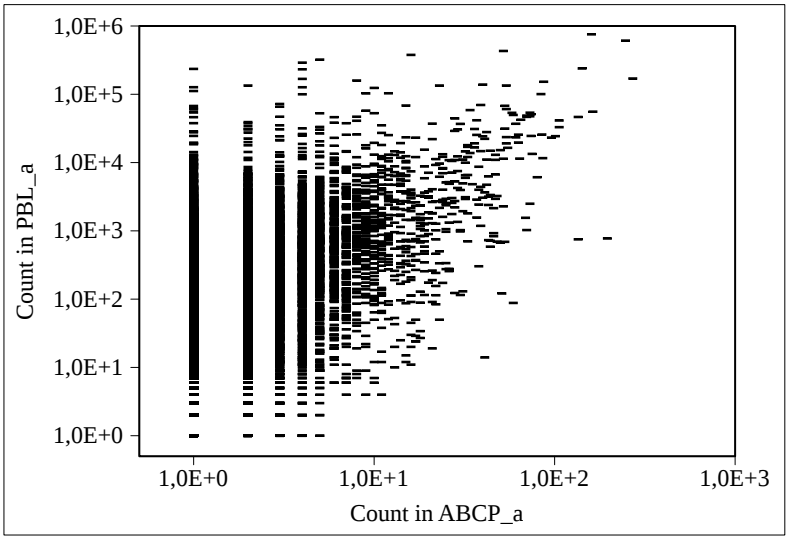

Figure 8: Plot of the counts of the (w_{-1}, c, w) occurrences in ABCP_a *versus* the counts in PBL_a.

somewhat frustrating) that some correlation between the counts in both sets becomes clear (even paying attention to the fact that log-scales are used) only when the counts in the ABCP_ data are higher.

In the Appendix E are given the necessary indications to have access to the code that was used to run the experiments here reported.

The main conclusion at this point is that the designed approach is efficient in the sense that is able to decrease the PP indicator in an independent data set. Follows that several limitations of the approach in the selection and adjusting steps, or at the designing and implementation levels, were identified, existing potential for substantial improvement (already exist some planning on these direction).

5 Future work

Several suggestions exist to proceed with this work. Some are more oriented to development issues and other are clearly oriented to research.

The following development-nature suggestions can be presented:

- try to implement *on-line* the *parts-of-speech* tagger that has been used (or other, eventually);
- improve automatically, using known procedures and the appropriate acoustic materials, the existing phonemic transcription of the words in the vocabulary (recall that the lexicons were obtained manually);

- improve automatically, following a similar approach based on the respective visual materials and, if possible, using a reliable syllabification tool, the existing *visyllabic* lexicon (recall that the LM2_LEX_VSL was build based on the LM1_LEX_VSL and not accounting for the visual realizations);

- improve the *gender* and *number* annotations, in the case of pursuing work in "hybrid grammars" (if confirms the perception that large imperfections exist, for sure that improvement would reflect greatly in the results);

- try to optimize parts of the code (C-language) which speed is more critical (in many cases, that was not a concern);

- get skills using LM tools other than the CMU-Cambridge LM-Toolkit (in particular, a brief incursion into the IRST LM Toolkit[10] impressed favorably).

In terms of research work, for the time being three main topics seem particularly attractive. They are related to work already initiated and reported in the Sections 4.3 and 4.4.

One of these topics, that is related with the LM_HG *hybrid grammars*, so addressing the problem of modeling linguistic knowledge beyond the words sequences statistics (in the case of the standard *ngrams*), certainly deserves additional research effort to:

- try improving the structure (eventually experimenting automatic structure learning techniques) and implementation aspects (for instance considering different *parts-of-speech* categories) of the model that uses jointly the POS and the gender and number inflection categories (there are a few ideas that could be tested, some quite easily though possibly bringing relatively modest gains);

- try adding knowledge to the model from other linguistic levels (for sure, semantic knowledge would be a candidate to allow substantial gains, even if the preliminary experiments had to be quite limited).

Another research topic here suggested addresses the language modeling robustness problem, in the perspective of the approach presented in the Section 4.4. To pursue that preliminary work, these two research lines could be followed:

- try designing techniques that could use applications prior knowledge and cues exhibited during the LM operation to select "candidates" to the *weak parameters* subset (for instance, in the used model, not necessarily implemented according to the equation 18, using approapriate

[10] `http://hlt.fbk.eu/en/irstlm`

heuristics combining prior knowledge with *on-line scores* on the occurrence of specific linguistic events, associated to variables in $P(w|w_{-1}, c)$, to get cues about potentially less robust parameter sets) [11];

- try designing efficient strategies and techniques to adjust only the selected parameters using de larger, though possibly with quite different linguistic characteristics, data (there are several ideas that could be experimented, in the particular case of the $P(w|w_{-1}, c)$ model).

Still regarding to the robustness, the third suggestion addresses the problem this time following the *smoothing* approach. In spite of the very superficial work referred in the Section 4.3.4, and also considering the profusion of specific existing techniques, there is the conviction that new efficient approaches can be developed. In particular, interesting research opportunities apparently exist, considering appropriate linguistic knowledge related to the application, in the *smoothing* approach framework (for instance, such simple ideas as try using efficiently knowledge related with the variable c, in the model $P(w|w_{-1}, c)$), to regulate the discount levels.

6 The conclusions

One of the main goals of this work, consisting of building the LM modules for the ABCP1 speech recognizer, was fully achieved. For most of the modules several versions are available. The following lexicons are ready to use jointly with the ABCP_CP1 corpus:

- LM1_LEX_v1 - single-pronunciation; phonetically based, not considering the phonetic context; with 7 599 entries; standard linear lexicon, to use in pass-1 (decoder);
- LM1_LEX_v2 - similar to LM1_LEX_v1 with the difference of using a lexical tree structure;
- LM2_LEX_VSL - single-pronunciation; *visyllabicaly* based, not considering the visual context; with approximately 642 entries; linear lexicon, to use in pass-2 (decoder);

And in respect to the grammars, for the ABCP_CP1 corpus are actually available:

- LM1_WP - *word-pair*, with 7K vocabulary;
- LM1_1G - *unigram* for the same text and vocabulary than LM1_WP;

[11] This initial step of the approach presented in the Section 4.4 was simplified in the experiments carried out, just considering the knowledge component.

- LM2_2G_v1 to LM2_2G_v6 - *bigrams*; with different versions, changing the text (subset used on training), the vocabulary, the smoothing method or the size; arpa format;

- LM2_3G_v1 and LM2_3G_v2 - *trigrams* for ABCP_CP1 corpus; with two versions changing the smoothing method; arpa format.

Globally, very positive results were obtained for another important goal, concerning the problem of encoding efficiently into the grammars knowledge related to the morpho-syntactic categories assigned to the words in the sentence, eventually considering knowledge on the concordance of gender, or number, inflections. Several probabilistic models were developed, that can be integrate into the ABCP1 recognizer if running *off-line* simulations. The code that was developed is available, such as the following two hybrid-grammars (with the sub-modules in the arpa format):

- LM2_HG_v1 - for the ABCP_CP1 corpus, based on the proposed model (equation 9);

- LM2_HG_v2 - for the PUBLICO corpus, based on approximation (equation 17) to the proposed model (and with three different sub-versions);

Approximately $17\%_{rel}$ decrease in the perplexity estimate in an independent set is obtained with LM2_HG_v2, taking as reference the result obtained with a baseline grammar. The work done allows to conclude that this topic offers appealing research opportunity.

Interesting results were also achieved with an approach that was designed, and partialy implemented, to the problem of reducing the effects of the lack of specific raining data. Using parsimoniously a large estra dataset, trying to adjust only some of the presumably less robust parameters, even in a very unfavorable context mainly due to the large mismatch between the original LM and that extra data, it was possible to reduce the perplexity in $1.47\%_{rel}$. Though modest in absolute terms, that result suggests this approach can be quite effective and advantageous in certain applications and contexts. Here were reported just the preliminary experiments already executed, and a much larger effort must be made to develop the idea. The code that was used to get these results is available (indications in the Appendix E).

Another initial goal was to study different approaches to the problem of using efficiently linguistic knowledge (for instance, based on POS tags) in the context of the *ngrams* smoothing. The work done with that objectif did not give the pretended results, yet. By the other side nothing was found that could disencourage to continue pursuing this objectif, on the contrary. By the way, it worth to mention the design and implementation of a simple (though quite easily allowing extensions to account for possible

new dependencies) variation based on the Witten-Bell discount technique, allowing to establish different discount levels.

It must be also referred the creation of a text corpus, ABCP_CP1, that although being small presents some interesting characteristics. In particular, this corpus has a suitable linguistic content, according to the speech recognition tasks that are intended to address, and integrates acoustic and visual annotation information on the respective audio-visual captured materials.

The following innovative aspects, for the best of the author's knowledge, can be assigned to the work here reported:

- the definition of the *visyllables* as the sub-word units at the basis of the recognizers visual model, offering an interesting compromise between the precision and the size of that model.

- the use of the *gender* and *number* inflections, jointly with standard *ngrams* already combined with *parts-of-speech* categorization knowledge, leading to substantial improvement in the modeling ability;

- the idea of adapting a non-robust existing model, based 1) on *prior* knowledge and *operation cues*, used to identify some subset of its *weak parameters*, 2) and on the existence of a relatively large dataset, which linguistic content possibly presents a large mismatch in relation to the existing model, 3) so that, then, an appropriate strategy could be implemented to adjust the selected parameters of the original model.

After the many experiments and readings along the work here reported, concerning to the author's interests in language modeling, in the context of the automatic speech recognition, the following achievements must be emphasized:

- the acquisition of *know-how* on many different tasks, in particular those related with building LM modules using existing tools;

- the *insight* gained on many topics (in general, the initial knowledge was very superficial), as distinct as those related to the models robustness (with acquisition of good intuition in this particular topic) or the use of less-conventional linguistic knowledge (with the intention to continue studying this subject).

References

[X. Huang (2001)] Xuedong Huang, Alex Acero, Hsiao-Wen Hon. Spoken Language Processing - A Guide to Theory, Algorithm, and System Development. Prentice Hall, 2001.

[A. Akmajian (2001)] Adrian- Akmajian, Richard A. Demers, Ann K. Farmer, and Robert M. Harnish. Linguistics, An Introduction to Language and Communication. The MIT Press, Cambridge - Massachusetts, London - England, 2001.

[H. Mateus (1999)] Maria Helena Mateus, Ana Maria Brito, Inês Duarte, and Isabel Hub Faria. Gramática da Língua Portuguesa. Editorial Caminho, 1999.

[Casa da Moeda (2009)] Imprensa Nacional - Casa da Moeda. Acordo Ortográfico da Língua Portuguesa. Editorial Caminho, Jan/2009.

[P. Ladefoged (1994)] P. Ladefoged. A course in phonetics, 3^{rd}ed. New York: H. B. Javanovich, 1994.

[S. Russel (2004)] Stuart Russel, Peter Norvig. Artificial Intelligence - A Modern Approach. Elsevier, 2004.

[J.-P. Tremblay (1984)] Jean-Paul Tremblay, Paul G. Sorenson. An Introduction to Data Structures with Applications. McGraw-Hill Intl. Editions,1984.

[H. Peng (2005)] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.8, August,2005.

[C. Ding (2005)] Chris Ding, and Hanchuan Peng. Minimal Redundancy Feature from Microarray Gene Expression Data. Journal of Bioinformatics and Computational Biology, Vol.3, No.2, pp.185-205,2005.

[M. Federico (2010)] Marcello Federico. Tutorial on Language Models. FBK-irst, Trento, Italy, 2010.

[Joshua Goodman (2002)] Joshua Goodman. The State of The Art in Language Modeling. In *presentation at the 6th Conf. of the Association for Machine Translation in the Americas (AMTA)*, Tiburon, CA, 2002.

[S. Chen (1996)] Stanley F. Chen. Building Probabilistic Models for Natural Language. PhD thesis, Harvard U., 1996.

[Katrin Kirchhoff (2008)] K. Kirchhoff, J. Bilmes, and K. Duh. Factored Language Models Tutorial, Tech. Report UWEETR-2007-0003, Dept. of EE, U. Washington, 2007.

[Dan Jurafsky (xx)] Dan Jurafsky. Language Modeling, Lecture 11 of his course on "Speech Recognition and Synthesis" at Stanford.

[C. V. Gasperin (2001)] Caroline V. Gasperin, and Vera L. Lima. Fundamentos do Processamento Estatístico da Linguagem Natural. Tech. Report, faculdade de Informática - PUCRS, Brazil, 2001.

[Goodman(2001)] Joshua T. Goodman. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research, 2001.

[Goodman(2001)] Joshua T. .Goodman. A bit of progress in language modeling. In *Computer Speech and Language (2001)15, 403-434*, 2001.

[Katz(1987)] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35(3):400–401, March 1987.

[Ian Witten (1991)] Ian H. Witten, and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. In *IEEE Transactions on Information Theory*, V. 37, N. 4, pp. 1085-1094, 1991.

[Issam Bazzi (2002)] Issam Bazzi. Modelling Out-of-Vocabulary Words for Robust Speech Recognition. PhD thesys, MIT, 2002.

[Kneser and Ney(1995)] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 181-184, 1995.

[Boulos Harb (2009)] Boulos Harb, Ciprian Chelba, Jeffrey Dean, and Sanjay Ghemawat. Back-off language model compression. In *Proc. of Interspeech*, pp. 325-355, Brighton, UK, 2009.

[A. Fernandes Jr. (xx)] Alcebiades Fernandes Jr.. Nível Morfológico, xx.

[J. L. Silva (1997)] Joo L. Silva. Utilizao do Paradigma Multi-Agentes no Processamento da Linguagem Natural. Dissertao de Mestrado, Instituto de Informática, U. Católica do Rio Grande do Sul, Porto Alegre, 1997.

[J. Gao (2005)] J. Gao, H. Suzuki. An Empirical Study on Language Model Adaptation. ACM Trans. on Asian Language Information Processing, Vol.5, N.3, September, 2005.

[J. Bellegarda (2001)] J. Bellegarda. An Overview of Statistical Language Model Adaptation. ITRW on Adaptation Methods for Speech Recognition, August 2001.

[M. Federico (2004)] M. Federico, N. Bertoldi. Broadcast news LM adaptation over time. Computer Speech & Language 18(4): 417-435, 2004.

[M. Federico (1999)] M. Federico. Efficient Language Model Adaptation Through MDI Estimation. In *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH '99)*, pp. 1583-1586, 1999.

[L. Chen (1999)] L. Chen and T. Huang. An Improved MAP Method for Language Model Adaptation. In *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH '99)*, pp. 1583-1586, 1999.

[M. Federico (1996)] M. Federico. Bayesian Estimation Methods for N Gram Language Model Adaptation. In *Proceedings of the International Conference on Spoken Language Processing*, pp. 240-243, 1996.

[H. Masataki (1997)] Task Adaptation Using MAP Estimation in N Gram Language Model. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 783-786, 1997.

[D. Okanohara (2007)] Daisuke Okanohara, and Jun'ichi Tsujii. A discriminative language model with pseudo-negative samples. Proc. of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 2007.

[E. Charniak (2003)] Eugene Charniak, Kevin Knight, and Kenji Yamada. Syntax-based Language Models for Statistical Machine Translation. In *MT Summit IX. Intl. Assoc. for Machine Translation*, 2003.

[Kuo (2002)] Hong-kwang J. Kuo, Eric Fosler-lussier, Hui Jiang, and Chinhui Lee. Discriminative training of language models for speech recognition. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, 2002.

[Brown et al.(1992)Brown, Pietra, DeSouza, Lai, and Mercer] P.F. Brown, V.J. Della Pietra, P.V. DeSouza, J.C. Lai, and R.L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[Elman(1990)] J.L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[F. Peng (2003)] Fuchun Peng, and Dale Schuurmans. Combining Naive Bayes and n-Gram Language Models for Text Classification. In *25th European Conf. on Information Retrieval Research (ECIR '03)*, 2003.

[S. Chen (1995)] Stanley F. Chen. Bayesian Grammar Induction for Language Modeling. 1995.

[J. Gao (2005)] Jianfeng Gao, Hao Yu, Wei Yuan, and Peng Xu. Minimum Sample Risk Methods for Language Modeling. In *Proc. of Human Language Technology Conf. on Empirical Methods in Natural Language Processing*, Vancouver, 2005.

[L. Saul (1997)] Lawrence Saul, and Fernando Pereira. Aggregate and Mixed-Order Markov models for Statistical Language Processing. 1997.

[M. Szarvas (2003)] Máté Szarvas, and Sadaoki Furui. Finite-State Transducer based Modeling of Morphosyntax with Applications to Hungarian LVCSR. In *Intl Conf on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol.1, pp.I, 368-371, 2003.

[K. Kirchhoff (2005)] Katrin Kirchooff, and Mei Yang. Improved language Modeling for Statistical Machine Translation. 2005.

[D. Z. Hakkani-Tur (2000)] Dilek Z. Hakkani-Tur, Kemal Oflazer, and Gokhan Tur. Statistical Morphological Disambiguation for Agglutinative Languages. Computers and the Humanities 36: pp. 381-410, 2000.

[M. Federico (2007)] M. Federico, M. Cettolo. Efficient Handling of N-gram Language Models for Statistical Machine Translation. In *ACL 2007 Workshop on Statistical Machine Translation*, pages 88–95,

[A. Cardenal-Lopez (1993)] A. Cardenal-Lopez, F. J. Diguez-Tirado, and C. Garcia-Mateo. Fast LM look-ahead for large vocabulary continuous speech recognition using perfect hashing. In *Intl Conf on Acoustics, Speech, and Signal Processing (ICASSP '93)*, vol.1, no., pp.I, 27-30, 1993.

[David Gutherie (2010)] David Guthrie, Mark Hepple, and Wei Liu. Efficient Minimal Perfect Hash Language Models. In *Proc. of the 7th Conf. on International Language Resources and Evaluation (LREC'10)*, La Valletta, Malta, 2010.

[J. Riedler (2003)] Jurgen Riedler, and Sergious Katsikas. 'My Small Slim Greek ASR System' or Automatic Speech Recognition of Moden Greek Broadcast News. In *Proc. of the Eurospeech 2003*, 2003.

[S. Chen (1998)] S. F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. Tech. Report TR-10-98, Computer Science Group Harvard U., Cambridge, August 1998 (original postscript document).

[S. Chen (1996)] Stanley F. Chen, and Joshua T. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proc. of the 34th Annual Meeting of the ACL (ACL '96)*, 1996.

[K. Baker (2004)] Kirk Baker. Constraining User Response via Multimodal Dialog Interface. International Journal of Speech Technology, 7, pp. 251-258, 2004

[P. Shinn (2004)] Phil Shinn, Matthew Shomphe, Molly Lewis, Kathy Carey, and David Kim. Designing Language Models for Voice Portal Applications. International Journal of Speech Technology, 7, pp. 93-99, 2004.

[H. Schmid(1994)] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. Proceedings of the International Conference on New Methods in Language Processing, pp. 44-49, 1994.

[H. Schmid(1995)] Helmut Schmid. Improvments in Part-of-Speech Tagging with and Application to German. Proceedings of the ACL SIGDAT-Workshop, pp. 47-50, 1995.

[M. Federico (2010)] M. Federico, N. Bertoldi and M. Cettolo. IRST Language Modeling Toolkit (Version 5.50.02) - User Manual. FBK-irst, Trento, Italy, 2010.

[N. Bertoldi (2008)] Nicola Bertoldi. A tutorial on the IRSTLM library. In *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH '99)*, pp. 1583-1586, 1999.

[P. Clarkson (1997)] P. Clarkson and R. Rosenfeld. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH '97)*, 1997.

[A. Stolcke (2002)] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, USA, 2002.

[S. Young (2006)] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland. The HTK Book (for HTK Version 3.4). Cambridge University Engineering Dept., 2006.

[V. Pera (2010)] V. Pera, "An Overview of the ABCP1 Speech Recognizer", Tech. Rep., ABCP, Porto, 2010.

[V. Pera (2011a)] V. Pera, "The ABCP-db1 Data Set", Tech. Rep., ABCP, Porto, 2011.

[V. Pera (2012)] V. Pera, "The Decoder of the ABCP1 System", Tech. Rep., ABCP, Porto, 2012.

A The IPA/*ABCP* symbols set mapping

IPA	*ABCP*
a	a
ɐ	6
ɐ̃	6∼
ɛ	E
ɛ̃	y
e	e
ẽ	e∼
ɘ	@
i	i
ĩ	i∼
ɔ	O
o	o
õ	o∼
u	u
ũ	u∼
j	j
w	w
w̃	w∼
b	b

IPA	*ABCP*
p	p
d	d
t	t
g	g
k	k
m	m
n	n
ɲ	J
v	v
f	f
z	z
s	s
ʒ	Z
ʃ	S
ʎ	L
l	l
ɬ	h
ɾ	r
R	R

Table 20: Conversion table between the IPA and the *ABCP* symbols sets.

B The ABCP_CP1 text corpus in brief

This small text corpus is associated to the ABCP-DB1 dataset, that was built to provide the audio-visual speech, and related text materials, needed to develop the ABCP1 recognizer. That text material comprises 8 524 daclarative sentences in the Portuguese language, with quite general linguistic scope, that were collected from a Brazilian source. The most notorious lexical differences detected, respecting to the european Portuguese, were corrected. Annotation information was extracted from that material using a morpho-syntactic analyser that is available in the Web (see Appendix D). More detailed information about this corpus can be found in the technical report[V. Pera (2011a)] respecting the ABCP-DB1.

The sentences of ABCP_CP1 are grouped in the set that in this report is denoted as ABCP, which was splet into two disjoint subsets, the ABCP_a and the ABCP_b. Table 21 has information concerning the size, both in terms of sentences or words, of these (sub)sets.

Number of ...	ABCP	ABCP_a	ABCP_b
sentences	8 524	6 524	2 000
words	78 723	59 479	19 244
different words	7 599	6 561	3 509

Table 21: Number of sentences and number of words and different words in the established sets.

C The CETEMPublico text corpus in brief

The text corpus referenced in this report by the name *Publico* consists of a set with more than 2 million sentences, selected from the CETEMPublico corpus, and annotation information extracted from it using a morpho-syntactic analyser. The CETEMPublico (Corpus de Extractos de Textos Electronicos MCT/Publico) is available in the Web address `http://www.linguateca.pt/CETEMPublico/`, from *Linguateca*. Contains approximately 180 M words, in the European Portuguese language, based on news and other journalistic text materials.

The sentences selected have in common the fact that all are declarative and, following an *ad-hoc* approach, were eliminated those presenting some sorts of peculiarities (such as non-portuguese words, acronyms, etc.). These sentences are grouped in the set denoted as PBL and three disjoint subsets were extracted from it: PBL_a, PBL_b and PBL_c. Table 22 contains some figures respecting these sets.

Set name	No. sentences	No. words
PBL	2 307 209	43 411 150
PBL_a	1 615 047	32 007 253
PBL_b	461 422	9 136 408
PBL_c	230 720	4 574 198

Table 22: Number of sentences and number of words in the established sets.

Table 23 presents several vocabularies that were defined and the respective sizes: VCB_all contains all the different words in PBL; VCB_10+ contains all the words in PBL that occur at least 10 times; VCB_10+_ABCP contains all the words in VCB_10+ that also exist in the ABCP sentences; and VCB_20K contains the 20K most frequent words in PBL.

Vocabulary	Size
VCB_all	224 598
VCB_10+	57 175
VCB_10+_ABCP	5 711
VCB_20K	20 000

Table 23: Size of different vocabularies associated to PBL.

The annotation information extracted from each (sub)set, using a morpho-syntactic analyser available in the Web (see Appendix D) is also available.

D The morpho-syntactic analyser and tagset

The morpho-syntactic analyser that was used is available in the Web addresses
`http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html`
and
`http://gramatica.usc.es/~gamallo/`.
In this last site can be found a paremeters set for the portuguese language, allowing promptly running the tagger. In both sites can be found the code and resources, or the respective links, needed to build other parameters sets, eventually considering different tagsets.

Tag	Comment
ADJ	gender & number
ADV	
CARD	
CONJ	
CONJSUB	*que* (clauses cojunction)
DET	gender & number
I	interjection
NOM	gender& number
P	gender & number ; pronoun
PR	relative pronoun
PRP	preposition
V	number
P+P	gender & number
PRP+ADV	
PRP+DET	gender & number
PRP+P	gender & number
X	(non-defined)

Table 24: PoS tags and related information.

The tagger is described in the articles "Probabilistic Part-of-Speech Tagging Using Decision Trees"[H. Schmid(1994)] and "Improvments in Part-of-Speech Tagging with and Application to German"[H. Schmid(1995)].

E The ABCP1 Language Model filesystem

The following files, containing the available resources of the ABCP1 language model ('tar' files contain code), can be found in the Web address `http://speech-rec-vcp.com/abcp1/lang-model/`

```
---------------------------
 LM1_LEX/lex-phn.dat
        /lex-phn-c.tar
 LM2_LEX_VSL/lex-vsl.dat
            /lex-vsl-c.tar
 LM1_WP/word-pair.dat
       /word-pair-c.tar
 LM1_1G/1gram.dat
       /1gram-c.tar
 LM2_2G/2gram-v1.arpa
       /2gram-v2.arpa
       /2gram-v3.arpa
       /2gram-v4.arpa
       /2gram-v5.arpa
       /2gram-v6.arpa
 LM2_3G/3gram-v1.arpa
       /3gram-v2.arpa
 LM2_HG/hyb-gram.tar
 LM2_ADPT/adapt.tar
---------------------------
```