

On the use of multigram models for automatic translation

Yassine Mami, Géraldine Damnati

France Télécom R&D - TECH/SSTP/RVA
2 Avenue Pierre Marzin - 22307 Lannion - FRANCE
{yassine.mami,geraldine.damnati}@francetelecom.com

Abstract

In this paper, we present a study and experiments on the feasibility of using multigrams for automatic translation. The multigram approach is based on a variable-length sequence modeling. First, we train a multigram sequences set and we translate them automatically using the bilingual training corpora. In the test step, each sentence is segmented into sequences of multigrams and compared to the reference sentence. This study was done in the framework of a pharmacy application, on an internal France Telecom database.

1. Introduction

The field of machine translation is almost as old as the modern digital computer. For years, because processors were not fast enough to handle the extensive computation these systems require, many experts considered statistical systems inferior to rules-based systems. Today, computers are five orders of magnitude faster than they were in 50s and have hundreds of millions of bytes of storage. On the other hand, the growing availability of bilingual, machine-readable texts has also stimulated interest in statistical methods for machine translation. Statistical methods have proven their value in automatic speech recognition and have recently been applied to natural language processing and machine translation. The goal is the translation of a text given in some language F into a target language E . The statistical translation approach (Brown et al., 1990) (Brown et al., 1993) (Ney, 1999) is based on the noisy channel model. Using Bayes rule, the translation probability for translating a source sentence f into a target sentence e is:

$$\arg \max_e p(e|f) = \arg \max_e p(f|e)p(e) \quad (1)$$

where $p(e)$ is the language model for the target language and $p(f|e)$ is the string translation model (cf. Figure 1). $p(f|e)$ refers to both a lexicon model and an alignment model. Extending the notion of lexicon from words to phrases has already proven to be helpful for Statistical Machine Translation. Phrases are defined from a bilingual aligned corpus as blocks that contain no gap and no overlap. The alignment issue is then decomposed into within phrase alignment and inter-phrase alignment.

In this paper, we present a study and experiments on the feasibility of extending the notion of phrases by using multigrams for automatic translation. The attempt of this study is to go further towards the definition of blocks that are “self contained” with regards to the alignment issue. The multigram approach is based on a variable-length sequence modeling and was introduced by (Bimbot et al., 1994) (Deligne and Bimbot, 1998) (Zitouni et al., 1998). In the multigram approach, it is assumed that a sentence is considered as the concatenation of independent variable-length sequences of words, and the likelihood of the sentence is computed as the sum of individual

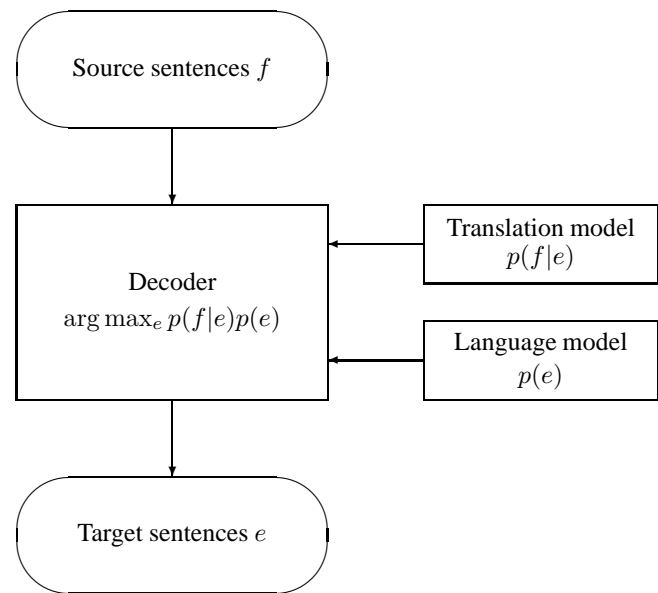


Figure 1: Statistical translation (Nießen et al., 1998).

likelihood of each possible segmentation. In practice, the Viterbi approximation is often made, replacing sum by max. In a first approach, in order to overcome the alignment model, only those multigram sequences that are connected from the alignment point of view are kept in the dictionary, that is only those sequences that can be translated into a connected sequence into the target language.

Our translation system, for the case of French to English translation, proceeds in two steps:

- Training step (figure 2).
 - From a French corpus, train multigram sequences (which represent the dictionary).
 - Discard sequences that are not connected from the alignment point of view.
 - Translate the remaining dictionary into English (translations are automatically derived from the aligned corpus)

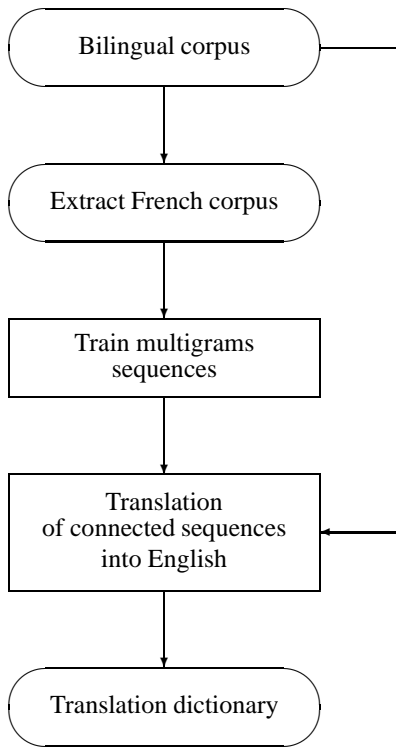


Figure 2: Training step.

- Testing step (figure 3). For each test sentence:
 - Find the most likely segmentation according a given dictionary.
 - Replace each segment of the sentence by its translation.
 - Compare translated sentence to the reference sentence.

This study was done in the framework of a pharmacy application, on an internal France Telecom database. The use case is an automatic translation system as a mediator between a pharmacist and a foreign customer. The paper is structured as follows: in the next section, we recall the multigram formulation. In section 3., we present the context of our application, the aligned corpus design and precise how the multigram sequences are translated. In section 4., we present the evaluation of the translation module and give some discussions on results.

2. Multigram formulation

In the multigram framework (or x-grams in opposition to n-grams), the assumption is made that sentences result from the concatenation of a variable-length phrases, called multigrams (Bimbot et al., 1995) (Deligne and Bimbot, 1995) (Deligne, 1996).

Let $W = \{w_1, \dots, w_t, \dots, w_T\}$ denote a string of words and let $S = \{s_1, \dots, s_q\}$ be a possible segmentation of W where each segment s_i has maximum length of m words. The joint likelihood of the sentence W and segmentation S

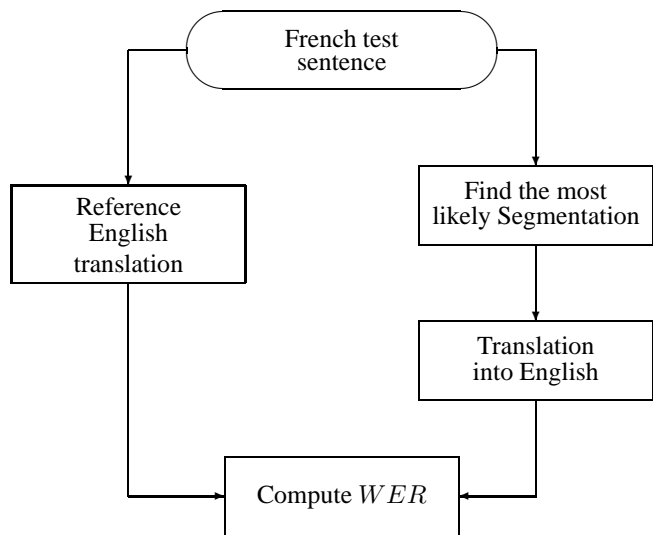


Figure 3: Testing step.

is the product of probabilities of each segment s_i :

$$\mathcal{L}(W, S) = \prod_{i=1}^{i=q} p(s_i) \quad (2)$$

In the translation framework, we need to use multigrams for two tasks, one in the training step and the other for the testing step:

- Parameter estimation: compute the multigram sequences probabilities.
- Segmentation: find the most likely segmentation of a sentence.

2.1. Parameter estimation

The estimation of multigram parameters $p(s_i)$ is obtained by maximum of likelihood. The likelihood of the sentence W is the sum of individual likelihood of each possible segmentations S :

$$\mathcal{L}(W) = \sum_S \mathcal{L}(W, S) \quad (3)$$

As the likelihood $\mathcal{L}(W)$ is a non linear function of multigram parameters, the direct maximization is quit difficult. EM algorithm can readily resolve this type of estimation problems (Dempster et al., 1977), where the observed data is the string of words W and the unknown data is the segmentation S underlying the string of words. The estimation of the model parameter is an iterative process which can be decomposed in three steps:

1. Initialize estimation of sequence probabilities (cf. paragraph 2.1.1.).
2. Reestimate the probabilities of multigram sequences (cf. paragraph 2.1.2.).
3. (2) is repeated iteratively until the likelihood converges or maximum number of iterations is reached.

2.1.1. Initialization

Let denote $\mathcal{D} = \{s_1, \dots, s_m\}$ the dictionary that contains all the multigram sequences s_i . This dictionary can be formed by combinations of 1, 2, \dots up to m words of the language vocabulary. The probabilities of sequences $p(s_i)$ can be initialized from the number of occurrences of a given sequence s_i :

$$p^0(s_i) = \frac{c^0(s_i)}{c^0} \quad (4)$$

where:

- $c^0(s_i)$ is the number of occurrences of segment s_i in the training corpora,
- c^0 is the number of total occurrences of all segments.

2.1.2. Parameter estimation

The probabilities of multigram sequences can be obtained by EM algorithm or Viterbi algorithm. In this article, we use EM algorithm to train parameters. The reestimation formulae of multigram sequences according to EM algorithm is the following:

$$p^{(k+1)}(s_i) = \frac{\sum_S c(s_i|S) \mathcal{L}^{(k)}(W, S)}{\sum_S c(S) \mathcal{L}^{(k)}(W, S)} \quad (5)$$

where:

- $\mathcal{L}^{(k)}(W, S)$ is the likelihood of string W at iteration (k) ,
- $c(s_i|S)$ is the number of occurrences of s_i in a given segmentation S ,
- $c(S)$ is the number of all sequences.

Equation 5 is implemented using the *forward-backward* formulas. For a given sequence s_i of l words, the reestimation formulae of multigram sequences is the following:

$$p^{(k+1)}(s_i) = \frac{\sum_{t=1}^T \alpha_l^{(k)}(t) \beta^{(k)}(t) \delta_{[w(t-l+1) \dots w(t)]}^{s_i}}{\beta^{(k)}(0) \gamma^{(k)}(T)} \quad (6)$$

where

$$\delta_{[w(t-l+1) \dots w(t)]}^{s_i} = \begin{cases} 1 & \text{if } [w(t-l+1) \dots w(t)] = s_i \\ 0 & \text{otherwise} \end{cases}$$

The *forward* variable $\alpha(t)$ is the likelihood of the partial string $W_1^t = \{w_1, \dots, w_t\}$. It is defined as:

$$\begin{aligned} \alpha(t) &= \sum_{l=1}^n \alpha(t-l) p([w(t-l+1) \dots w(t)]) \\ &= \sum_{l=1}^n \alpha_l(t) \end{aligned} \quad (7)$$

The second *forward* variable $\gamma(t)$ represents the average number of sequences in a possible segmentation of $W_1^t = \{w_1, \dots, w_t\}$:

$$\gamma(t) = 1 + \sum_{l=1}^n \gamma(t-l) \frac{\alpha_l(t)}{\alpha(t)} \quad (8)$$

The *backward* variable $\beta(t)$ is also needed, it represents the likelihood of the last $(T-t)$ words of the string W :

$$\beta(t) = \sum_{l=1}^n p([w(t+1) \dots w(t+l)]) \beta(t+l) \quad (9)$$

with $\alpha(0) = \beta(T) = 1$, $\gamma(0) = 0$ and $1 \leq t < T$.

2.2. Segmentation

The most likely segmentation is obtained by maximum of likelihood:

$$\hat{S}(W) = \arg \max_S \mathcal{L}(S, W) \quad (10)$$

There are many techniques to find the most likely segmentation. In this article, we use the Viterbi algorithm which avoid the explicit search of all segmentations.

At each iteration (k) , the algorithm proceeds as follow:

- Initialization

$$\begin{cases} \delta(0) = 1 \\ \psi(0) = 0 \end{cases}$$
- Recursion

$$\begin{cases} \delta(t) = \max_{l=1, \dots, n} \delta(t-l) p(w_{t-l+1} \dots w_t) \\ \psi(t) = t - \arg \max_{l=1, \dots, n} \delta(t-l) p(w_{t-l+1} \dots w_t) \end{cases}$$
- Termination

$$\mathcal{L}^*(W) = \max_S \mathcal{L}(W, S) = \psi(T)$$

3. Multigram sequences for speech translation

3.1. Experimental context and corpus generation

This study was done in the framework of a pharmacy application. Typically, a French tourist, not speaking English, who travels in an Anglophone country. In the case where this tourist has some health problems, he can go to the pharmacy to ask some medicines or some advises. The study aims to build an automatic translation system which facilitates communication between the chemist and the tourist. For bootstrapping purpose, a bilingual corpus has been created by a professional interpreter and manually aligned at the word level.

The database has been structured in terms of situations that can be combined following several scenarios. A sub-corpus has been created for each situation. For a given situation, sentences have been built by making use of *a priori* classes (e.g. for medicines, body parts,...). In order to facilitate the translation and alignment steps for elements inside classes, we did not restrict the elements of classes to content words but we introduced articles and prepositions and therefore duplicated the classes depending on the context. For instance the word *paracétamol* can appear in the class `_MEDICINE1_` with the formulation *du paracétamol*, in the class `_MEDICINE2_` with the formulation *au paracétamol*, in the class `_MEDICINE3_` with the formulation *de paracétamol*, etc... Following this principle, a set of 92 classes has been designed.

From a set of generating grammars, it is potentially possible to extract a large amount of sentences but in order to

avoid redundancy we have selected a set of 4680 sentences. Among these sentences, we selected randomly 4000 sentences for training and 680 for test corpora.

In order to train the multigram model, we chose to remain at the class level. The alignment has been achieved with a relative position framework. The French sentences have been manually segmented into word sequences, in order to obtain minimal elements that can be assigned a corresponding element in the English sentence. Here is an example of segmented and aligned pair of sentences, each containing 8 elementary segments.

- *je ne me sens pas /0/ bien /0/ je /0/ me /+1/ suis coupé /-1/ _BODY_PART1_/0/ hier /0/ matin /0/*
- *I'm not feeling / well / I / cut / my / _BODY_PART1_ / yesterday / morning /*

3.2. Multigram training and sequence validation

The elementary segments constitute the initial vocabulary for training the multigram sequences. The training corpus, at the class level, contains 491 different elements for an overall amount of 38750 occurrences. The multigram parameters were trained by EM algorithm (see equation 5). To assign a non-zero probability to unseen multigram sequences, we trained multigram parameters on corpora composed of the training corpus and the vocabulary. The performances in term of perplexity seem slightly better than trained multigram parameters on only the training sentences.

In order to validate the sequence connectivity, we added a verification step to guarantee that a multigram sequence can translate into a sequence that does't contain any gap or overlap. For the moment, sequences that do not verify the connectivity property are simply discarded from the multigram dictionary. This validation step has been achieved by simply verifying that the relative positions of a given multigram sequence sum up to zero. For instance, the sequence *suis coupé _BODY_PART1_* would be discarded because the corresponding position indexes sum up to -1 where as the sequence *me suis coupé _BODY_PART1_* because the indexes sum up to zero. A translation of this latter sequence can be derived as *cut my _BODY_PART1_*.

3.3. Translation dictionary

The translation of the dictionary is carried out as shown figure 5. For each multigram sequence S among the dictionary, we find all French training sentences W_{french} which contain S . Then, for each sentence W_{french} containing the multigram sequence S , we verify if the connectivity property described in the previous section is respected. If so, we find the translation of S using the aligned pair sentences $\{W_{french}, W_{english}\}$: we joint the aligned words of each words of S , according to their positions and alignment orders.

For each sentence which contains the multigram sequence S , we obtain a translation of this sequence. So, for all sentences, we can obtain several translations. There are various ways to select one translation among all. Thus, we can:

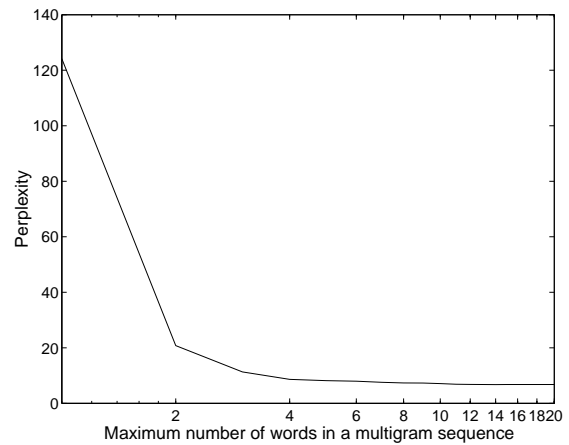


Figure 4: Perplexity variations according length sequences.

- select the most frequent translation
- validate the translation by a language model of the target language
- select a translation by combining the scores of the two items (frequency and language model).

In this study, we consider only the most probable translation, that is the one which is the most frequent in the training corpus.

4. Evaluation

4.1. Language model evaluation

If we assume that the language model is an independent entity, the perplexity is the most widely used evaluation metric for language models. It is given by the following equation:

$$PP(W) = 2^{-\frac{1}{T} \log_2 \mathcal{L}(W)} \quad (11)$$

The perplexity indicates the degree of difficulty to predict the language by the probabilistic model. The smaller perplexity is, the better language model is.

4.2. Optimization of multigram parameters

Before evaluating the translation system, we first optimize different parameters of multigram sequences. So, we measure the perplexity according to the following parameters:

- The maximum number of words in a multigram sequence m .
- The number of occurrences above which a sequence of words is included in the initial inventory of sequences.
- The number of iterations in training step.

Figure 4 shows a plot of the perplexity for sequences which occur m times or less and for minimal number of occurrences equal to 3. We have tested different values of the number of iterations in training step. For a fixed value of the maximum number of words in a multigram sequence and a fixed number of the minimal number of occurrences,

the perplexity reached a minimum value for 10 iterations of training algorithm.

Figure 4 shows that the perplexity is maximal for an unigram model (each sequence is composed of a single word). Then, it decreases rapidly for a maximum number of words in a multigram sequence less than 5. Beyond 5 words in a sequence, the perplexity stay relatively stable and the minimum number of occurrences seems not a critical a parameter for perplexity.

Our translation system is based on multigram approach with the maximum number of words in a sequence is equal to 5.

4.3. Translation dictionary

The multigram sequences obtained after training constitute the dictionary for translation. The number of units in the dictionary changes according the maximum number of words in a multigram sequence. Table 1 gives the number of units for different multigram models.

Model	Number of units
multigram $m = 2$	952
multigram $m = 3$	1058
multigram $m = 4$	1122
multigram $m = 5$	1223
multigram $m = 6$	1257

Table 1: Number of units in the dictionary.

As shown in table 1, we have 1223 multigram sequences to translate into English. The dictionary contains 490 unigram, 128 sequence of 2 words, 164 sequences of 3 words, 181 sequences of 4 words and 260 sequences of 5 words. So, the average sequences length is 2.7 words. Here some examples of the most likely ones:

- *pourriez-vous me donner,*
- *vais prendre une boîte de,*
- *s'il-vous-plaît,*
- *...*, etc.

4.4. Evaluation of our translation system

As shown in figure 3, the test procedure can be decomposed in three steps:

- For each sentence W_{test} , find the most likely segmentation \hat{S} by using Viterbi algorithm (see section 2.2.).
- Replace each multigram sequence of W_{test} by its English translation. The translation of whole W_{test} is the concatenation of each sequence translation.
- Compare reference translation and those of multigrams.

Evaluation of the quality of any translation is quit difficult, since it is not entirely clear what the focus of the evaluation should be. Surely, a good translation has to adequately capture the meaning of the foreign original. However, many

translation systems use the BLEU metric (Papineni et al., 2002).

To evaluate our translation system, we aligned the translation obtained by multigram approach and the human translation. Then, we compute the alignment errors exactly in the same way as we do it in a continuous speech recognition system. Thus, the word error rate is given by the following formulae:

$$WER = \frac{Ins + Sub + Del}{OK + Sub + Del} \quad (12)$$

where Ins , Sub et Omi represents respectively the number of iterations, the number of substitutions and the number of deletions. OK is the number of words correctly recognized.

Table 2 summarize the word translation performances. We obtained good performances, about 94.7% of correct rate and $WER = 7.0\%$. This can be explained by the translation task which is relatively easy with this corpus. Indeed, the French training corpus is artificial and has been created by generating grammars. Although these experiments are preliminary, they show that the multigram approach, combined with a connectivity validation is promising. The fact of simply concatenating the translations of sequences, assuming that the word order changes are restricted to intra-sequence moves, shows that multigrams capture very relevant sequences.

Percent Correct	94.7%
Word Error Rate	7.0%
Percent Substitution	3.1%
Percent Deletions	2.2%
Percent Insertions	1.7%

Table 2: Word translation performances.

A large part of errors come from the fact that a word or a sentence can be translated into several ways. In the bilingual corpora, we can find different translations for the same sentence. For example, the sentence “*je ne me sens pas bien*” is translated into:

- *I'm not feeling well*
- *I've not been feeling well*
- *I've been feeling off colour*

Same comments for the sentence “*très bien*” which is translated into:

- *very well*
- *very good*
- *fine*

5. Conclusion

In this paper, we present a study and experiments on the feasibility use of the multigrams for automatic translation. The

preliminary experiences on the pharmacy application, show that we obtain good performances in term of word error rate. The translation task on this type of corpus is relatively easy. That's why further experiments should be done on the transcription of a real spontaneous corpora. The most important result is that the multigram approach captures very relevant sequences, which can be used in automatic translation.

Further study will be focused on the integration of the word position model into the training step of the multigram sequences and on the use of a target language model. We'll be also study the use of joint multigram (Deligne et al., 1995) which seem to be well-suited for the automatic translation task.

6. Acknowledgements

We are grateful to Sabine Deligne and Frédéric Bimbot for supplying multigram toolkit (Deligne and Bimbot, 1997).

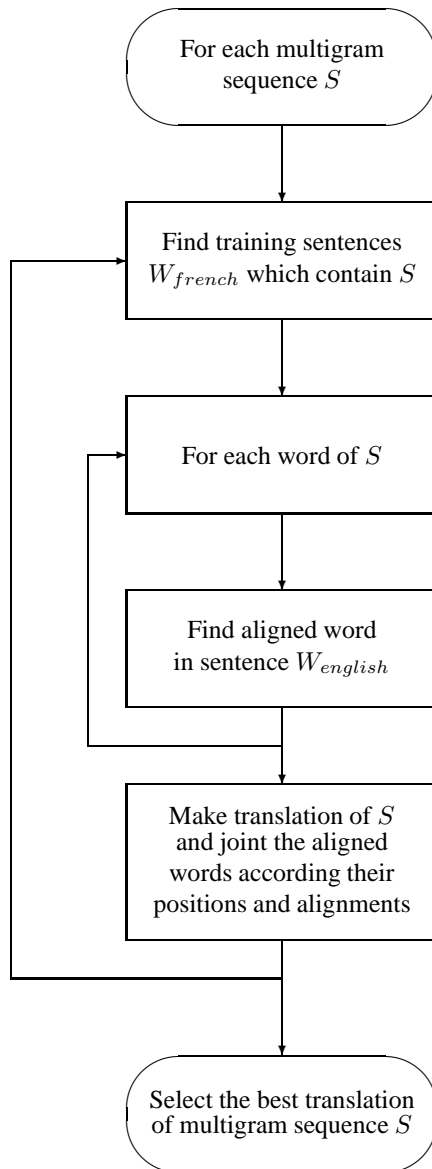


Figure 5: Sequences translation.

7. References

- F. Bimbot, R. Pieraccini, E. Levin, and B. Atal. 1994. Modèles de séquences à horizon variable : Multigrams. In *Journées d'Etude sur la Parole*.
- F. Bimbot, R. Pieraccini, and B. Atal. 1995. Variable-length sequence modeling: Multigrams. In *IEEE Signal Processing Letters*, volume 2, pages 111–113.
- Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederik Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.
- P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.
- S. Deligne and F. Bimbot. 1995. Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 169–172.
- S. Deligne and F. Bimbot, 1997. *Multigram Package*. Ecole Nationale Supérieure des Télécommunications - Paris.
- S. Deligne and F. Bimbot. 1998. Learning a syntagmatic and paradigmatic structure from language data with a bi-multigram model. In *International Conference on Computational Linguistics*, pages 300–306.
- S. Deligne, F. Yvon, and F. Bimbot. 1995. Variable-length sequence matching for phonetic transcription using joint multigrams. In *European Conference on Speech Communication (EUROSPEECH)*, pages 169–172.
- S. Deligne. 1996. *Modèles de séquences de longueurs variables, application au traitement du langage naturel et de la parole*. Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications - Paris.
- A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38.
- Hermann Ney. 1999. Speech translation: Coupling of recognition and translation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, page 1675.
- S. Nießen, S. Vogel, H. Ney, , and C. Tillmann. 1998. A dp based search algorithm for statistical machine translation. In *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 960–967.
- K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.
- I. Zitouni, K. Smaili, J.-P. Haton, S. Deligne, and F. Bimbot. 1998. A comparative study between polyclass and multiclass models. In *International Conference on Spoken Language Processing (ICSLP)*.