Learning for Sequence Extraction Tasks

Massih-Reza Amini, Hugo Zaragoza, Patrick Gallinari
LIP6, University of Paris 6,
case 169, 4 Place Jussieu
75252 Paris cedex 05, France
{amini, zaragoza, gallinari}@poleia.lip6.fr

Abstract

We consider the application of machine learning techniques for sequence modeling to Information Retrieval (IR) and surface Information Extraction (IE) tasks. We introduce a generic sequence model and show how it can be used for dealing with different closed-query tasks. Taking into account the sequential nature of texts allows for a finer analysis than what is usually done in IR with static text representations. The task we are focusing on is the retrieval and labeling of texts passages, also known as highlighting and surface information extraction. We describe different implementations of our model based on Hidden Markov Models and Neural Networks. Experiments are performed using the MUC6 corpus from the information extraction community.

1. Introduction

With the increase of electronically available textual information, new needs for Information Access systems are arising. Many new tasks lie between the classic frameworks of Information Retrieval (IR) and Information Extraction (IE). Machine Learning (ML) is playing a central role in the development of these fields but has been used for the most part for the improvement of existing models.

We propose to extend the capabilities of statistical IR models to handle more complex information retrieval and extraction tasks. For this, we explore the use of probabilistic sequence models for sequence analysis. In particular, we will consider a text as a sequence of symbols and not as an unordered set. From this perspective, a sequence model is proposed that permits to work at a finer level than what is usually done in IR. This model is capable of dealing with several text analysis tasks with a unifying formalism.

For representing word sequences, we propose a term mapping onto a very low dimensional space, this mapping has been found surprisingly efficient for relatively complex tasks. We then introduce the sequence model which allows to handle different closed query tasks, ranging from filtering and routing to the detection of relevant phrases in texts. Our model has been implemented using neural networks and hidden Markov models. By *closed-query tasks* we denote those tasks where the information need is known in advance so that this knowledge can be used to construct the system, as opposed for example to the ad-hoc IR task.

In this paper, we focus on two example applications of the sequence model: *passage highlighting* and *surface information extraction.* In the first, we wish to select the most pertinent sequences of words within a document, with respect to a specific task or information interest. While, in the second, we wish to label words according to several sub-interests.

We use Wall Street Journal articles from the MUC-6 corpus (MUC 6, 1996) and associated Scenario Templates to test our models. In this context, we wish to highlight all the descriptions of *personnel change* events (job appointments, reassignments or job terminations) and, furthermore, label the *name* and *position* of the person concerned.

In section 2 we review related work in the ML, IR and IE communities. The probabilistic framework of our approach is introduced in section 3, we discuss text representation in section 4 and derive the probabilistic sequential models in section 5. Section 6 is devoted to evaluation, we first show that the proposed representation effectively captures the information needed for the different tasks and then illustrate and analyze the behavior of our models on highlighting and surface extraction tasks. Comparisons are provided with respect to baseline classifiers.

2. Related work

Machine learning has been used for several years in IR. Since IR mainly relies on statistical tools, any numerical ML model, classifier or ML density estimation technique can be used in place of common statistical techniques for closed-query tasks. Decision Trees (Koller & Sahami, 1997), Neural Networks (Schutze et al., 1995; Wiener et al., 1995), Support Vector Machines (Joachims, 1998) have been used for filtering and routing tasks. Classifier comparisons for text categorization have been performed by several authors (Dumais et al., 1998). Note that all these models consider a bag of words text representation.

In IE there have been also several attempts over the last years to use machine learning algorithms in order to automate the different processing steps of an IE chain. For the extraction step, Autoslog (Rilof, 1993) has been one of the very first systems using a simple form of learning, successors to autoslog like CRYSTAL (Soderland et al., 1995) mainly use decision trees and relational learning techniques for learning sets of rules during their extraction step. More recently, the SrV system (Freitag, 1998) uses a combination of relational and basic statistical methods inspired from Naïve Bayes for information extraction tasks.

Sequence models, mainly HMMs have been recently proposed for handling different tasks in IR or IE. (Mittendorf & Schauble, 1994) have been the first to propose a generative model based on HMMs for document retrieval and highlighting. Recently, (Miller et al., 1999) applied a similar approach - each document is modeled by an HMM - for the ad-hoc task on the large document collections of TREC6 and TREC7.

In the field of IE, HMMs have been used for named entity extraction by (Bikel et al., 1999), their experiments show that simple ergodic models where each state is associated with a topic reach surprisingly good performances. (Leek, 1997) uses HMMs for extracting information in the form of simple binary relations between two entities on a limited domain in biology, this model has been carefully handcrafted for this task.

The work which is closest to ours is probably (Freitag & McCallum, 1999). Like we do, they consider the extraction of entities in the context of closed queries. They build a discrete HMM to extract information, each state representing a particular label. (Wermter et al., 1999) use recurrent neural networks – which are another formalism for sequence modeling - for routing.

Text segmentation has also been considered from an IR perspective. (Turney, 1999) uses decision trees for the extraction of keywords or keyphrases (two or more words) from text, framing the extraction problem as a classification problem on pre-segmented text. A large amount of work has also been dedicated to text segmentation into coherent passages, e.g. (Hearst & Plaunt, 1993; Ponte & Croft, 1997; Beeferman et al., 1999).

Compared to these works, the originality of our approach lies in the representation we are using for text encoding and in the development of a new generic model, which allows to handle several tasks within the same formalism. Other works using HMMs for IR and IE rely on term density estimation. Compared to this text representation, an advantage of our formalism is that the word encoded sequences may be naturally extended to include continuous or discrete information sources (section

4), and allow us to consider terms in context (section 6.2). Furthermore, output sequences may be naturally constrained to follow a set of constraints specified by a grammar (section 6.2).

3. Model description

We consider a *document d* as the realization of a random variable *D*. A document can be represented in a number of ways; in particular, a document can be considered at different levels of granularity, and coded with different functions. We consider a document *d* as a *sequence of words* encoded into a sequence of vectors representing individual terms, $w=\{w_1,...,w_n\}$. The choice of a particular coding function will be discussed in Section 5. *d* may have an associated *document class*, *c*, which is a realization of the random variable *C*.

Furthermore we will consider that a term sequence *w* may have an associated *label sequence t*. Label sequences are realizations of the random variable *T*, the set of labels being denoted $\{\tau_1,...,\tau_K\}$. When we want to extract different classes of information from the text *w*, the *t* sequence will indicate the term labels (type or class of each word). For routing tasks, *t* indicates the relevancy of terms with respect to an information need. These different levels of description are indicated in Figure 1.

Although paragraphs and phrases may be treated with the vector space or probabilistic models of IR, these models are not naturally fitted for handling this information. This is however necessary to perform complex information access tasks. In this paper we will focus on two such tasks: highlighting, which is an intermediate complexity task, and surface IE which is a minimal message understanding task. The proposed sequence model handles these tasks naturally since it considers term sequences.

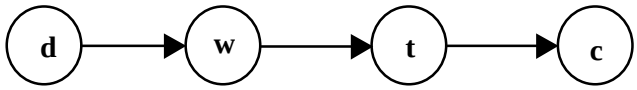

Figure 1. The different types of information associated with a document: *d* is the document raw sequence of words, *w* is the corresponding encoded term sequence, *t* is a sequence of labels associated with terms in *w*, *c* is the class of a document. Note that only some of these information may be relevant for a given task.

Depending on the task, some information will be available to us, and some will be missing or irrelevant (see Figure 2). A number of closed-query tasks can be formulated within this formalism. To illustrate this point, let us consider the following four tasks:

Classification : we want to assign a class label to each document. For training, we dispose of a set of documents and their corresponding classes.

Ranking: we wish to rank a set of incoming documents with respect to their relevancy to a given class. We dispose of a set of documents and their corresponding classes for training.

Surface IE: We wish to analyze a document and label words that respond to particular concepts. For training we dispose of a set of labeled documents.

Section Highlighting: We wish to highlight sections of an incoming document that are most relevant to the different classes. We dispose of a set of documents and their corresponding classes for training, but we do not dispose of highlighted examples.

In Figure 2 we represent these tasks with respect to the general model of Figure 1. Thick circles represent the information known (the documents, in all cases) and dotted circles indicate the information that is missing or desired; filled in gray is the information available for training the model.

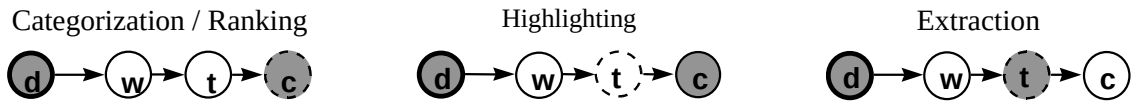

Figure 2: Closed-Query Tasks and the probability model considered. Thick circles represent the information known (the documents, in all cases) and dotted circles indicate the information that is missing or desired, and filled in gray is the information available training.

Within this framework, we will consider the evaluation of $P(w,t,c)$[1]. For any three variables w, t and c, $P(w,t,c) = P(w)\ P(t|w)\ P(c|t,w)$. A conditional independence hypothesis allows to drop the dependency of two terms, given the knowledge of a third. In the following we are going to assume that document class is conditionally independent of a document's word sequence representation, given the label sequence: $P(c|w,t)=P(c|t)$. This leads to the following joint probability distributions:

$$P(w,t,c) = P(c|t)\ P(t|w)\ P(w) \quad (1)$$

This decomposition is typical of sequence analysis, where a pipeline of process is used to analyze a sequence of tokens. From these distributions, we may express the tasks previously described.

In ranking, we are interested in the *class relevancy score* of a document d given a class c :

$$RSV(d,c) = P(c|w) = \Sigma_t P(c|t)P(t|w) \quad (2)$$

For classification, we are not interested in the score but on the class c* that maximizes it:

$$c^* = argmax_c \Sigma_t P(c|t)P(t|w) \quad (3)$$

where $argmax_c$ indicates the class c which maximizes the expression. Equations (2) and (3) extend the classical vector space and probabilistic IR models in two ways: *i)* the first term of the summation allows for complex relations between the information within a document and the document's class, and *ii)* the second term allows a sequential treatment of terms and labels.

The introduction of a "hidden" t sequence may seem artificial here, but it allows us to unify the treatment of the different tasks and label passages, e.g. one may wish to rank documents according to their most significant passage or with respect to an information extraction task.

When highlighting for categorization, we are interested in the most probable sequence of labels as well as on the class of the document. From (1), we have:

$$argmax\ P(c,t|w) = argmax_{c,t} P(c|t)P(t|w) \quad (4)$$

When highlighting a ranked list of documents, c is fixed and the distribution of interest is therefore:

$$argmax_t\ P(t|w,c) = argmax_t P(c|t)P(t|w)\ P(w)/P(w,c) = argmax_t P(c|t)P(t|w) \quad (5)$$

Finally, in Information Extraction, the class of a document is of no relevance, we look for

$$\hat{t} = \arg\max_t\ P(t,w) \quad (6)$$

The problem that needs to be solved is twofold: *i)* the determination of the probability distribution $P(c|t)$ (or the label *grammar* employed), and *ii)* the estimation of $P(t|w)$. The Viterbi algorithm (Rabiner & Juang, 1993) can be used for the fast computation of the maximization involved in equations (4-6).

[1] While it is possible to consider several coding functions through the use of the random variable D, we will consider in our present work that a single determinist function is used for the encoding of the document, thus ignoring the d term.

4. Sequence representation

Sequence models of the kind proposed here pose the problem of the dimensionality of sequence representation, since they relax the usual word-independence assumption (Bikel et al. 1999; Freitag & McCallum, 1999), using discrete class-conditionally independent term probabilities.

We proceeded differently: not wishing to consider class-conditionally independent term probabilities, we looked for a mapping from the term space to a very low dimensionality continuous space, which allows us to consider term dependencies within the sequence.

Such an approach has been previously proposed by (Mittendorf & Schauble, 1994) for open queries. Compared to the direct estimation of term distributions, they offer several advantages *i)* a more robust estimation (because of the reduced dimensionality), *ii)* the possibility to use continuous models, and *iii)* considering additional input information and local term context (section 6). Since we are dealing with persistent information needs, we may exploit in the coding function the fact that training documents are labeled. We will use here one such mapping, based on the U-measure (Anderson, 1992) which we have found efficient.

4.1 Continuous coding

Let us describe a word by the four values of the contingency table describing the presence or absence of a term in relevant and irrelevant contexts. Specifically, let *n* and *n'* denote respectively the number of relevant and irrelevant contexts in which a term appears. Let *m* and *m'* denote respectively the number of relevant and irrelevant contexts in which the term does not appear. Using these values, we can code a term as a four dimensional real valued vector. When *t* is available for training (like in information extraction tasks), the relevant contexts are phrases. When *t* is unknown and only *c* is known (e. g. for ranking), a context is considered as a full document.

We are thus mapping terms onto a 4 dimensional space, and distances in this new representation space will be correlated to patterns of term usage with respect to the extraction task. The semantics of this representation (and more elaborate forms of this approach) seem appropriate for surface information extraction. This representation can be interpreted as a weak domain-dependant semantic representation; words that have the same semantic function with respect to a given task will be mapped to nearby points in space.

We found that better results were obtained if we mapped the four values obtained onto the real line, using a discriminant measure to describe the contingency table. The well known χ^2 measure, for example, which indicates the discriminative power of the presence or absence of a term with respect to the class of relevant and irrelevant passages can be used. While the absence of a term may be important in IR applications, it does not seem an appropriate indicator for IE. We found the U-statistic (Anderson, 1992) more appropriate, since it only rewards positive correlation. This measure was proposed for feature selection in a routing application in (Knaus et al., 1994). For a given topic, the U-measure of the term w_i is defined as :

$$u_i = U(w_i) = \sqrt{N}.\frac{nm' - n'm}{\sqrt{(n+n')(n+m)(n'+m')(m+m')}} \qquad (7)$$

where *N* is the total number of phrases ($N=n+n'+m+m'$). Note that we use this measure to represent all the terms over the real line, not to select a subset of terms (Zaragoza & Gallinari, 1998). While this approach is surely simplistic, it has yielded promising results (see section 6).

We have also considered the use of morpho-syntactic information for the encoding process. Syntactic labels may dissociate some of the terms having similar U-values and thus contribute to a better representation of words for information extraction tasks, where the syntactic role of words is

important. Thus, we augmented the previously described feature space of terms with morpho-syntactic tags. We used for that a probabilistic Part-Of-Speech tagger[2] (Schmid, 1994).

4.2 Variable Selection

Since all features are not equally informative, we performed automatic selection on the feature set (Amini et al., 1999*b*). We have chosen here a method proposed by (Bonnlander et al., 1996) to remove all non informative features.

The relevance of a set of input variables is defined as the mutual information between these variables and the corresponding classes. This dependence measure is well suited for measuring non linear dependencies as they are captured in Neural Networks. For two variables x and d, it is defined as: $MI(x,d) = \sum_{x,d} P(x,d)\log\frac{P(x,d)}{P(x)P(d)}$. Here x denotes the encoding of a word w and d indicates the class associated to x (i.e. *I*, *Per* and *Pos* for the extraction task and *I* or *R* for highlighting).

Starting from an empty set, variables are added one at a time, the variable x_i selected at step i being the one which maximizes $MI(Sv_{i-1},x_i,d)$ where Sv_{i-1} is the set of $i-1$ already selected variables and d the desired output. Selection was stopped when the ratio $\rho = MI(Sv_{i-1}, d) / MI(Sv_i, d)$ raises above a fixed threshold (0.99 in our case).

We performed variable selection for the two tasks. In both cases, we obtained the same set of 5 variables: *U, PN, N, ADJ, V*. Table 1, gives for each step, the variable selected, $MI(Sv_i, d)$ and ρ, for the task of highlighting. In the following, we will use the two representations *U* and *(U, PN, N, ADJ, V)* as input to our models for highlighting and extraction.

Step i	Variable x_i	$MI(Sv_i, d)$	ρ
1	*U*	0.1779	
2	*PN*	0.1871	0.9506
3	*N*	0.1943	0.9633
4	*ADJ*	0.2009	0.9667
5	*V*	0.2067	0.9719
6	*DET*	0.2084	0.992
7	*O*	0.2092	0.9962
8	*CC*	0.2092	1

Table 1. Variable selection for highlighting

5. Estimating probabilities

5.1 Estimation of *P(c|t)*

Remember that $P(c|t)$ is the probability of the *document's class* given a sequence of *labels*. This term will not be estimated from data in general. Most often, this knowledge is not available. However, we can use this term to embed *a priori* knowledge of the domain. Specifically, it allows us to constrain the solution space and find sequences of certain user-defined characteristics. A natural way to express knowledge of document class given word classes is through finite state stochastic grammars (Rabiner & Jung, 1993; Zaragoza, 1999) (see section 6).

5.2 Estimation of *P(t|w)*

Sequence models translate a known input sequence $w_{1,n}= w_1w_2\ldots w_n$, $w_i\in \boldsymbol{W}$ of symbols into an output sequence $t_{1,n}=t_1t_2\ldots t_n$, $t_i\in \boldsymbol{T}$, where $\boldsymbol{W}$ and $\boldsymbol{T}$ are input and output spaces for the model, n the length of these sequences and w_i and t_i denote respectively the i^{th} element of the input and output sequence.

[2] http://www2.ims.uni-stuttgart.de/Tools/DecisionTreeTagger.html

There exists a joint probability distribution over all the possible input and output sequences. This probability is unknown and must be inferred from a set of labeled examples. In our case, the input symbol sequence is the encoded text itself and the output sequence is the sequence of labels that encode the information contained in the text. w_i is then the vector representation of the i^{th} word of the sequence and t_i its label.

Classically, the joint probability can be decomposed into a product of conditional probabilities. For extraction and highlighting we will consider the most probable labeled sequence associated to a sequence of words, i.e.

$$\hat{t}_{1,n} = \arg\max_{t_{1,n}} P(t_{1,n}, w_{1,n}) = \arg\max_{t_{1,n}} \prod_{i=1..n} P(t_i / t_{1,i-1}, w_{1,i}).P(w_i / t_{1,i-1} w_{1,i-1}) \quad (8)$$

Following (Charniak et al., 1993), we will derive two main expressions from (8). Under the following two assumptions of locality and independence, $P(w_i / t_{1,i-1}, w_{1,i-1}) = P(w_i / w_{i-1})$ and $P(t_i / t_{1,i-1}, w_{1,i}) = P(t_i / w_{i-k,i+k})$ equation (8) reduces to:

$$\hat{t}_{1,n} = \arg\max_{t_{1,n}} \prod_{i=1..n} P(t_i / w_{i-k,i+k}) \quad (9)$$

(8) can also be decomposed as:

$$\hat{t}_{1,n} = \arg\max_{t_{1,n}} \prod_{i=1..n} P(w_i / w_{1,i-1}, t_{1,i}).P(t_i / w_{1,i-1} t_{1,i-1})$$

And under the assumptions, $P(t_i / w_{1,i-1}, t_{1,i-1}) = P(t_i / t_{i-1})$ and $P(w_i / w_{1,i-1}, t_{1,i}) = P(w_i / t_i)$ it comes:

$$\hat{t}_{1,n} = \arg\max_{t_{1,n}} \prod_{i=1..n} P(w_i / t_i).P(t_i / t_{i-1}) \quad (10)$$

Expressions (9) and (10) correspond to two different models for decoding the best sequence of labels. The hypotheses they rely on are motivated by changing the initial combinatorial decoding problem into a more tractable one. Neural Networks and Hidden Markov Models allow to implement equations (9) and (10) respectively. Neural Network models can estimate both discrete and continuous probability distributions. HMMs can estimate continuous distribution probabilities using gaussian mixtures trained with the EM algorithm

Instead of (9), we have been using here a slightly more powerful model (equation 11) where transition probabilities $P(t_i / t_{i-1})$ have been added to (9)

$$\hat{t}_{1,n} = \arg\max_{t_{1,n}} \prod_{i=1..n} P(t_i / w_{i-k,i+k}).P(t_i / t_{i-1}) \quad (11)$$

To evaluate these two sequential models we have also implemented a set of discrete HMM models similar to those proposed in (Freitag & McCallum, 1999). These models operate on a discrete space, they do not take as input a continuous representation such as the one proposed in section 4, but a sequence of terms *w* corresponding to a simple preprocessing of the document word sequence (eliminating stop words and stemming). This model requires the estimation of $P(w_i / C_j)$ for each term w_i and topic C_j. We have used the following smoothed estimate :

$$\hat{P}(w_i / C_j) = \frac{tf(w_i / C_j) + 1}{N + \sum_{k=1}^{N} tf(w_i / C_k)} \quad (12)$$

Where, $tf(w_i / C_j)$ is the number of occurrences of w_i for the topic C_j, and N the number of topics. This estimator is suggested in (Vapnik, 1982), it assumes that the observation of each term is *a priori* equally likely. The Viterbi algorithm is then used to find efficiently the optimal state sequence .

6. Experiments

In order to evaluate our models we use the MUC-6 corpus (MUC 6, 1996) which has been developed by the IE community for the evaluation of IE systems. This corpus consists of a set of 200 Wall Street journal articles. For each document, a set of Scenario Templates (ST) describe instances of personnel changes (appointments, job terminations, etc.) (Figure 4). STs are subdivided into fields that describe a particular aspect of the event (Figure 4, bottom left). In the present work we concentrate on only two fields: the *Name* and the *Position* of the person concerned.

We designed a procedure to label automatically each word in the corpus with a class label. Since each document has an associated set of STs, and documents contain paragraphs delimiters, we compare each paragraph to each one of the document's ST. If the paragraph partially matches the contents of at least two ST fields, the paragraph is considered as relevant. Paragraphs are then subdivided into phrases (sequences of at least two words separated by punctuation marks or conjunctions). If a phrase contains the text of an ST field, all the words in that phrase are given the class of the ST field. All the other words are labeled as being irrelevant. This heuristic leads to some labeling errors, insertions and replacements; furthermore, it makes the assumption that STs are filled from text coming from a single paragraph, which is the case for most of them.

This labeling procedure makes some mistakes on the real labeling of each sequence of terms. We consider this bias as noise in data. Our goal is to provide models capable of handling labeling errors typical of automatic labeling systems.

The two example applications we are dealing with are passage highlighting and surface information extraction. In the first, we wish to highlight all the descriptions of personnel change events (job appointment, reassignment or job terminations), in which case our goal will be to label sub-sequences of text as Irrelevant (I) or Relevant (R) for the task (term labels take only two values 0/1). While in the second, we aim to extract the name and position of the person concerned, in which case sub-sequences are to be labeled as Person (Per), Position (Pos) or Irrelevant (term labels may take 3 values) (Figure 4, top right).

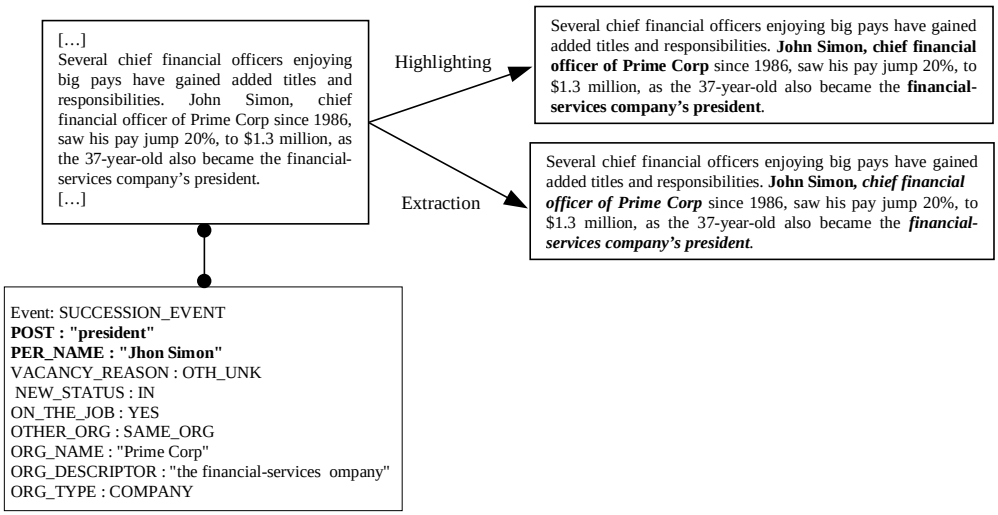

Figure 4: A paragraph (top left) is compared with a Scenario Template (bottom left) to obtain word labels (top right). In the Scenario Template box we have indicated in bold the two fields of interest: POST and PER_NAME. For the extraction task PERSON labeled words are shown in bold and POSITION labeled words are shown in italics. For the highlighting task both PERSON and POSITION labeled terms are shown in bold. IRRELEVANT words are shown in normal face.

This approach to labeling databases can be adopted in other situations besides MUC data. There are many common situations where we have access to a database containing textual fields, and a corpus of documents from which the information proceeds or can be extracted.

6.1 Representation

To demonstrate the representation power of the U-measure, we show in Figure 5, for a routing experiment, a precision and recall curve obtained by i) a Naïve Bayes classifier, and ii) using for the relevancy score of a document the mean U-measure of its terms[3].

For ranking, the U-measure can be directly used as a term weighting function and behaves very well compared to this Naïve Bayes classifier (Figure 5). We will see in the following section that this representations is also useful to carry out difficult tasks such as Information Extraction or highlighting.

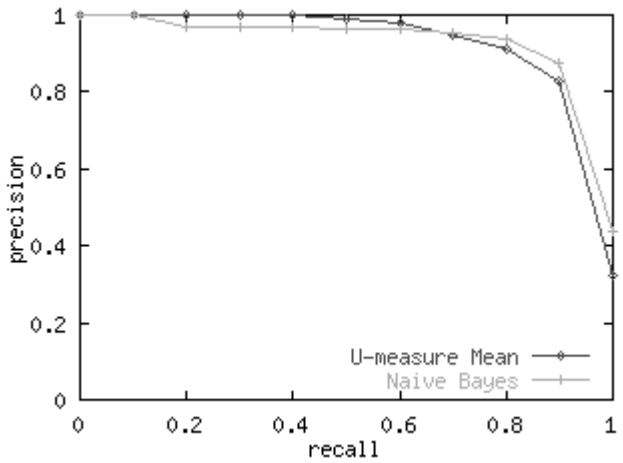

Figure 5. Precision-Recall curves for text categorization for two classifier systems, a naïve Bayes classifier and the mean U-measure of the text terms.

Another advantage of using low dimensional continuous representation is the possibility of using posteriori probability estimators, such as proposed in (2). We have performed different evaluations for routing using (2) where the conditional probability *P(t/w)* has been computed with multi-layer perceptrons (Bishop, 1995). The model behaves well, but in general, has only similar performances to simpler classical models like e.g. Naïve Bayes while being computationally heavier.

For difficult retrieval tasks (on short documents), however model (2) has been found superior to probabilistic models which rely on the estimation of term densities *P(w/c)*. Let us consider for example a routing experiment using paragraphs of the MUC-6 corpus. This corpus contains a total of 1671 paragraphs, of which 215 (12.8%) are relevant. Documents (i.e. paragraphs) are considered relevant if they contain any information necessary to fill the Scenario Templates of the MUC-6 information retrieval task.

The major difference between retrieving paragraphs and whole documents is in the ratio of relevant to irrelevant documents. Furthermore, relevant terms appear often throughout the entire document, while only one or two paragraphs are in general truly relevant (i.e. they describe an instance of a change of job position). Finally, the length of paragraphs is more uniform than that of documents in general.

---

[3] Results are given on a random test set of 6000 documents, obtained on the 20-newsgroup corpus (`http://www.cs.cmu.edu/~mccallum/bow/rainbow`). Each curve is the average of the 20 precision-recall curves obtained for the 20 classes (Usenet groups) of this corpus. U-measures and probabilities were obtained from the remaining 14000 documents. The RAINBOW system , publically available at the above URL, was used to implement the Naive Bayes model.

Our model is compared in Figure 6 to the Naive Bayes model. Both systems use the same preprocessing, which consists of eliminating a list of 200 stop words, stemming and a low frequency cut-off of three occurrences.

In order to evaluate the two models, a single random split of one third of the paragraphs was put aside into the test set. Full leave-one-out cross-validation was carried out with one of the models and the error observed was similar to that of the test-set.

For this task, the sequence model performs significantly better. The sequence model proves to be more discriminant, since it directly approximates $P(c|w) = \Sigma_t P(c|t)P(t|w)$, offering better estimates than probabilistic models (such as Naive Bayes) which compute estimates of $P(w/c)$ for ranking. Furthermore, the continuous low-dimensional mapping reduces the difficulty of the estimation problem.

Note however that the main interest of this model is clearly not in categorization since simpler models do as well, but in the additional possibilities it offers to the user. For example, highlighting while ranking (equation 4) or taking into account, while ranking, a grammar which specifies which type of information is of importance for ranking, etc.

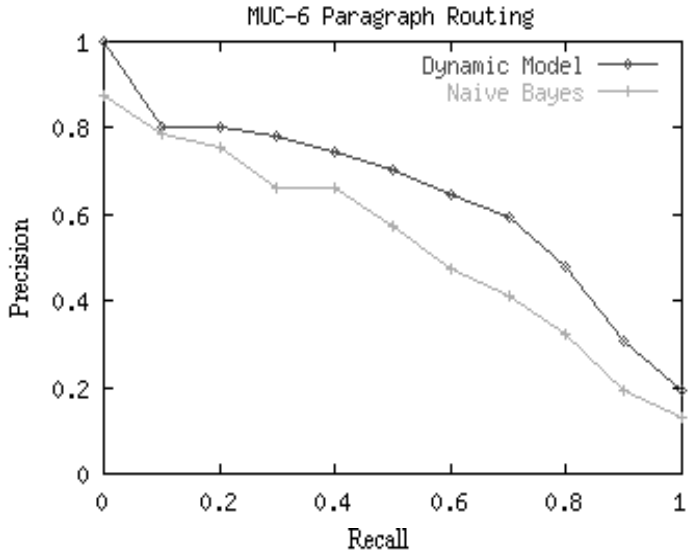

Figure 6 : Precision – Recall curves for the proposed sequence model and the Naive Bayes model, on the task of routing MUC-6 paragraphs.

6.2 Extraction and Highlighting

For highlighting and extraction, we have implemented the two models (10) and (11) respectively with HMMs for estimating $P(w_i/t_i)$ and $P(t_i/t_{i-1})$ and with MLPs for estimating $P(t/w)$. With our text representation, (11) is easily implemented by simply redefining the input of the network to be a window of words $w_i' = (w_{i-k}, ..., w_i, ..., w_{i+k})$. Note that this would be very difficult to do with the discrete term representation since it would amount to computing *n*-gram distributions (for $n>3$).

In both cases, a *grammar* may be used to constrain the allowed state transitions. As input to these models we have used the U-measure plus the morpho-syntactic tagging described in section 4. It has been shown (Amini et al., 1999*a*) that this representation led to better performances than the U-measure alone. We assume that all word occurrences fall into exactly one category, in the case of highlighting the two categories are r*elevant* or *irrelevant* and in the case of extraction *person*, *position* and *irrelevant*.

For comparison, we have also implemented the discrete baseline HMM (eq. 12) described in section 5.2. For the three models, text segmentation into the different categories was computed via the Viterbi algorithm.

The corpus was split into a training set and a test set. Altogether there were 100 and 105 paragraphs respectively in the training and the test set. About 8% of words are Person, 32% Position and 60% irrelevant.

Grammars

For both highlighting and extraction, the models can be forced to output constrained sequences, the constraint corresponding to a priori knowledge on the task or on the type of information the user is looking for. We have performed tests with different constraints.

We used for highlighting the approach proposed in (Mittendorf & Schauble, 1994) which consists in using a grammar of the type *I-R-I*, i.e. models are forced to tag the text as an Irrelevant *(I)* passage followed by a relevant (*R)* passage and again an irrelevant *(I)* passage.

Such models find the most probable single sub-sequence of relevant terms within the sequence. While this does not correspond truly to the way relevant information is present in the corpus, it provides a single résumé of a whole text which may be desirable. It can also be considered as a first approximation to our highlighting paradigm.

We also used the grammar $<I \mid R^{(3)}>$ where words may be labeled alternatively as I or *R*, (<> indicates one or more occurrences and | means or) but *R* sequences are forced to a minimal duration of 3. The latter has been chosen to introduce into the model the knowledge about the characteristics of the corpus.

Similarly, for extraction, we used grammars of the type *I-*<*Per-Pos-I*>, and $<I|Per^{(3)}|Pos^{(3)}>$ corresponding to a very simple configuration of a three state HMM shown in Figure 7.

The first and the second states show the probability distribution of terms in regard to the Person (*Per*) and the Position (*Pos*) class, the third state represents choosing a word from “general English” and so Irrelevant (*I*) for our extraction tasks.

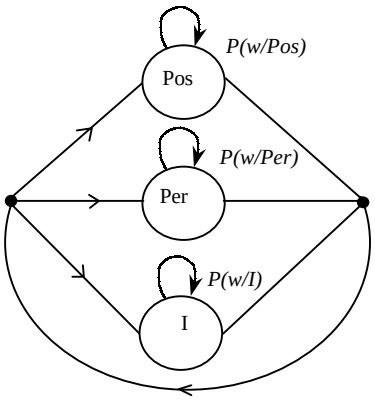

Figure 7. HMM model corresponding to the $<I|Per^{(3)}|Pos^{(3)}>$ topology forced to for the extraction task (duration model not shown). States have a probability distribution of terms in regard to each class, Person (Per), Position (Pos) and Irrelevant (I).

Evaluation

Table 2 gives the correct classification performances for the three models: (10), (11) and the baseline HMM of section 5.2. They are denoted respectively U_{MLP}, U_{HMM} and W_{HMM} . In a first experiment we used the *I-R-I* and *I-<Per-Pos-I>* grammars respectively for highlighting and for extraction. U_{MLP} and U_{HMM} models give a net improvement over the W_{HMM} model but it is not clear from these experiments which of these two models should be preferred. Average performances are however clearly superior for U_{MLP}.

	Highlighting		**Extraction**		
	% correct classification		% correct classification		
	Relevant	*Average*	*Position*	*Person*	*Average*
	I-R-I		*I-<Per-Pos-I>*		
U_{MLP}	81.06	**82.45**	61.58	**38.34**	**70.20**
U_{HMM}	**85.15**	62.73	**87.49**	12.29	57.37
W_{HMM}	45.29	60.02	55.49	10.31	56.34
	$<I\|R^{(3)}>$		$<I\|Per^{(3)}\|Pos^{(3)}>$		
U_{MLP}	85.32	**78.9**	74.31	**48.09**	**73.51**
U_{HMM}	**86.57**	71.02	**75.49**	22.31	66.17
W_{HMM}	41.13	67.52	43.44	0	65.69

Table 2: Performances of the U_{MLP}, U_{HMM} and W_{HMM} models for highlighting and surface extraction for two different grammars. In the case of highlighting, average is over Relevant and Irrelevant classes while in the case of extraction it is over Position, Person and Irrelevant classes. Best performances are in bold.

In a second experiment, we have performed the same tests as before with a minimal duration constraint. For highlighting, the performances of *I* and *R* are in a reversed order compared to the first experiment. For the extraction, the $<I|Per^{(3)}|Pos^{(3)}>$ grammar leads to a very neat increase in the performance: the percentage of correct labels for the different classes is well balanced while the global performances are the best which have been obtained in our tests. This shows that the model exploits successfully the introduction of domain knowledge.

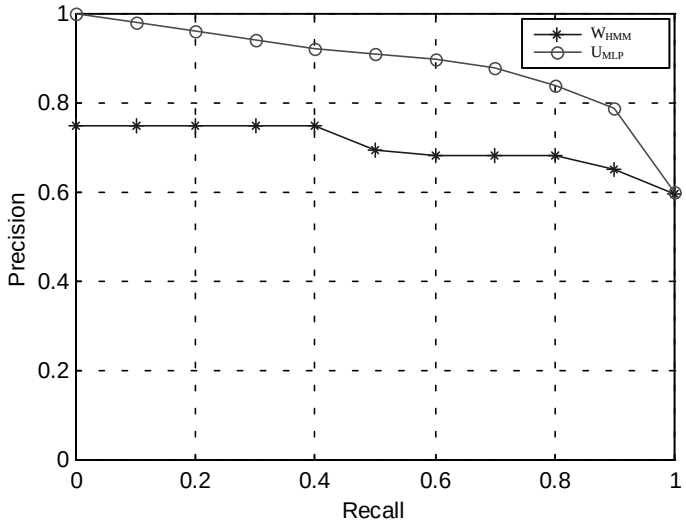

Figure 8: Precision-Recall curves obtained by the U_{MLP} (top line) and W_{HMM} (bottom line) models for highlighting.

The Performances for the Person class are much lower than those for Position for any model. The Person label has fewer occurrences (8%) than the Position label (32%) and they are most often classified as Positions (in one case for W_{HMM} all Person occurrences are classified as Position).

For all the models, transition probabilities greatly influence the performances and may change the balance between Person and Position (these probabilities could be optimally set via cross validation, but it has not been done here). In all experiments best mean performances are obtained with the continuous posterior estimator U_{MLP}.

Figure 8 shows precision-recall curves for U_{MLP} and W_{HMM} in the case of highlighting using the duration grammar $<I|R^{(3)}>$. Precision is defined as c_k/M and recall as c_k/N_k where c_k is the number of correctly classified words in the class *k (topic k)*, *M* the total number of words (correctly or incorrectly) judged relevant by the system and N_k the total number of words in the class *k.* All words are sorted by this relevancy, labeling the highest *M* as relevant and the rest as irrelevant.

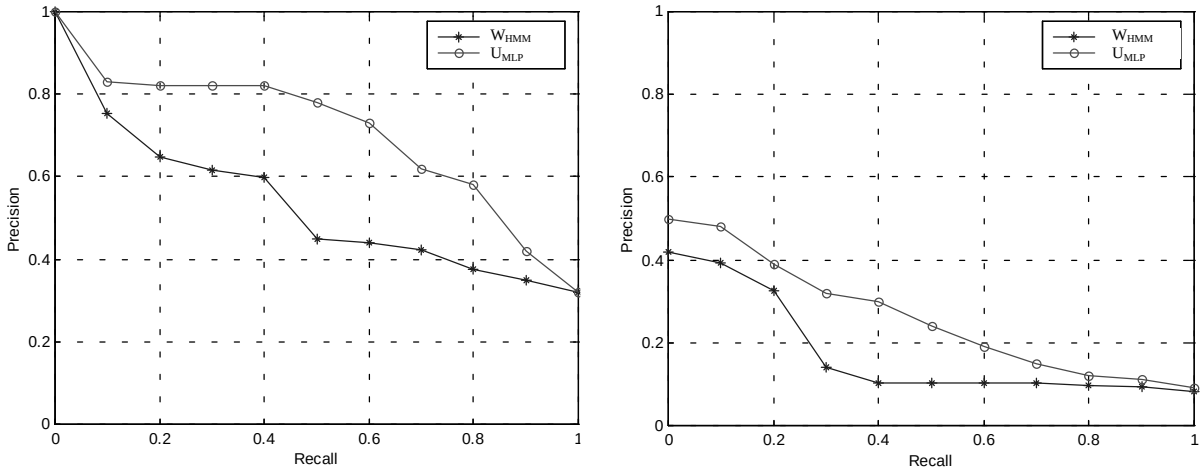

Figure 9: Precision-Recall curves obtained by the U_{MLP} and W_{HMM} models for extraction of Position class (left) and Person class (right)

In Figure 9, the same experiment is illustrated in the case of extraction. The precision-recall curves are obtained for both Position (Figure 9 left) and Person (Figure 9 right) classes. In all experiments, the U_{MLP} model clearly outperforms the baseline discrete HMM. For both models, performances for the Person class are rather low.

7. Conclusion and Perspectives

We have described a unified formalism for the automatic analysis of word sequences in the context of closed-query Information Access tasks. These formalism is quite general and derives into different probabilistic models depending on the characteristics of the information need. Such models operate on a low dimensional representation of word sequences. They can be implemented using continuous maximum likelihood estimators (ex. HMMs) as well as posterior probability estimators (ex. MLPs). Further, label grammars are introduced into the formalism to take into account domain dependant knowledge.

We have illustrated their potential on two application examples: surface Information Extraction and Highlighting. Their behavior and performances have been compared to a baseline discrete HMM models. For each of these two tasks we have shown the interest of using label grammars.
Furthermore, we have shown that the term representation using the U-measure embeds superficial semantic information. Using this measure as a weighting function leads to a better routing results than

the Naive-Bayes model for the 20-newsgroup corpus. For a difficult paragraph routing task (MUC-6 paragraphs), a sequence model using this measure outperformed the Naive-Bayes model.

Our sequence models are still perfectible, we are currently investigating new implementations for document ranking and phrase extraction and new applications for automatic résumé and text structure extraction. Although grammars has been shown to influence considerably model behavior, their role for the different text analysis tasks needs to be further studied.

8. Bibliographical References

Amini M.-R., Zaragoza H., Gallinari P. (1999)*a*, Sequence Models For Automatic Highlighting and Surface Information Extraction. *Information Retrieval Special Group, BCG-IRSG'99,* (pp. 85--91).

Amini M.-R., Zaragoza H., Gallinari P. (1999)*b*, Stochastic Models for Surface Information Extraction in Texts, *International Conference on Artificial Neural Networks, ICANN'99*, (pp.892--897).

Anderson E. (1992), *The Statistical Analysis of Categorical Data*, Berlin : Springer.

Beeferman D., Berger A., Lafferty J. (1999), Statistical Models for Text Segmentation, *Machine learning, ML'99*, 34.

Bikel D., Schwartz R., Weischedel R.M. (1999) An algorithm that Learns what's in a Name, *Machine learning, ML'99*, 34, (pp. 211--231).

Bishop C.M., (1995), *Neural networks for Pattern Recognition*, Oxford : Clarendon Press.

Bonnlander, Weigend A.S. (1996), Selecting Input Variables Using Mutual Information and Nonparametric Density Evaluation, *in Proceedings of the International Symposium on Artificial Neural Networks, ISANN'96,* (pp. 42--50).

Charniak E., Hendrickson C., Jacobson N., Perkowitz M. (1993), Equations for part of speech tagging, *11th Nat. Conf. On AI, AAAI*, (pp.784--789), MIT Press.

Dumais, S. Platt, J. Heckerman, D. Sahami, M. (1998), Inductive Learning Algorithms and Representations for Text Categorization. *In Proceedings of the 7th International Conference on Information and Knowledge Management, ICIKM'98.*

Freitag D. (1998), Machine Learning for Information Extraction in Informal Domains, PhD Thesis, Carnegie Mellon University, C.S. Department.

Freitag D., McCallum A. (1999), Information extraction with HMMs and shrinkage, *AAAI-99 workshop on machine learning for information extraction.*

Hearst M.A., Plaunt C. (1993), "Subtopic structuring for full document access", SIGIR 93.

Joachims T. (1998), Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *European Conference on Machine Learning, ECML'98*, (pp. 137--142).

Knaus D., Mittendorf E., Schauble P. and Paraic S. (1994) Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System. *In Proc. of the 4th Text REtrieval Conference, TREC4.*

Kohonen T., Kasaki S., Lagus K., Honkela T. (1996) Very Large Two-Level SOM for the Browsing of NewsGroups. *International Conference on Artificial Neural Networks, ICANN'96*, LNCS, (pp.269--274).

Koller, D., Sahami, M., (1997) Hierarchically Classifying Documents using Very few Words, *Proc. of the 14th International Conference on Machine Learning ICML'97*, (pp. 170--178).

Leek T.R. (1997), Information Extraction using Hidden Markov Models, Master thesis, University of California, San Diego.

Miller, D.R.H. Leek, T. Schwartz R.M. (1999), BBN at TREC7: using hidden Markov Models for Information retrieval, *Proceedings of TREC7*, D.K. Harman, editor.

Mittendorf E. and Schauble P. (1994), Document Passage Retrieval Based on Hidden Markov Models, *ACM SIGIR'94*, (pp. 318--327).

MUC 6, (1996), *Proceedings of the sixth Message Understanding Conference*, Morgan Kaufmann, Publishers.

Ponte J.M., Croft W.B. (1997), Text segmentation by topic, *in Proceedings of the First European Conference on Research and Advanced Technology for Digitial Libraries*, (pp. 120--129).

Rabiner L., Juang B.H. (1993) *Fundamentals of Speech Recognition*, Prentice Hall Signal Processing Series.

Rilof E. (1993), Automatically Constructing a Dictionary for Information Extraction Tasks, *AAAI'93*, (pp. 811--816).

Salton, G. and McGill, M. J. (1983) *Introduction to Modern Information Retrieval*. McGraw-Hill Publishers, New York.

Schmid H. (1994) Probabilistic Part-of-Speech Tagging Using Decision Trees. *In Proc. of Int. Conference on New Methods in Language Processing*, Manchester, UK.

Schutze H., Hull D. and Pederson J. O. (1995) A Comparaison of Classifiers and Document Representations for the Routing Problem, *In Proc. 18th International Conference on R&D in IR (SIGIR)*, (pp. 229--237).

Seymore K., McCallum A., Rosenfeld R. (1999) Learning hidden Markov model Structure for Information Extraction, *AAAI-99 workshop on machine learning for information extraction.*

Soderland S., Fisher D., Aseltine J., Lehnert W. (1995), CRYSTAL: Inducing a Conceptual Dictionary, *in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI'95*, (pp. 1314--1319).

Turney P. D., (1999), Learning to extract keyphrases from text. *Technical Report ERB-1057, National Research Council, Institute for Information Technology*.

Vapnik V., (1982) *Estimation of Dependencies Based on Empirical Data*, Springer-Verlag Publishers.

Wermter S., Arevian G., Panchev C. (1999), Recurrent neural networks for text routing, *International Conference on Neural Networks, ICANN'99*, (pp. 898--903).

Wiener E., Pedersen J.O., Weigend A.S. (1995), A Neural Network Approach to Topic Spotting, *in Proceedings of the 4th Symposium on Document Analysis and Information Retrieval, SDAIR'95*, (pp. 317--332), Las Vegas, NV, USA

Zaragoza H. (1999), Modèles Dynamiques d'Apprentissage Numériques Pour l'accès à l'Information, Thèse de Doctorat en Informatique, Université de Pierre et Marie Curie, Paris 6.

Zaragoza H. and Gallinari P., (1998), Coupled Hierarchical IR and Stochastic Models for Surface Information Extraction, *Information Retrieval Special Group, BCG-IRSG'98*, (pp.198--206).