Paper presented at the: *20th Annual Colloquium on IR Research, British Computer Society's Information Retrieval Specialist Group, (IRSG'98), Grenoble 1998.* Available at: `http://www-poleia.lip6.fr/~zaragoza`

Coupled Hierarchical IR and Stochastic Models for Surface Information Extraction.

Hugo Zaragoza, Patrick Gallinari.

LIP6, *Université Pierre et Marie Curie*,
4, place Jussieu F-75252,
PARIS cedex 05 (F).
{Hugo.Zaragoza, Patrick.Gallinari}@lip6.fr

Abstract

We present in this paper a combination of Machine Learning based Information Retrieval (IR) techniques and stochastic language modelling in a hierarchical system that extracts surface information from text. At the lowest level of this hierarchy, documents and paragraphs are successively *routed* with IR techniques. At the top level, a stochastic language model extracts the most relevant phrases, and labels the type of information they contain. The approach and preliminary results are demonstrated on a subset of the MUC-6 Scenario Templates task.

1 Introduction

The extraction of information in textual data, today, relies mainly on linguistic analysis. Successful systems have been demonstrated for many information extraction problems [7]. These systems are usually complex and expensive to build, aimed at limited domains of knowledge and difficult to extend. In order to overcome these problems Information Extraction (IE) systems recently began to integrate machine learning (ML) techniques [5]. The strength of ML techniques is their capacity to create automatically models that fit the data and adapt to new data. Their weakness, is their inability to handle complex relations. However several categorization and extraction tasks, where the information is implicit in the data, can already be handled by these methods; they can then be viewed as complementary or sometimes alternative methods to more classical IE techniques.

ML techniques have already established themselves as a successful alternative to hand-crafted linguistic oriented approaches in the domain of information extraction in automatic speech recognition (ASR). In this domain, we are usually interested in assigning *conceptual tags* to words or word sequences, either for translating this sequence into a request for information in some formal language, or in anticipating conceptual classes to improve recognition [15, 19]. Similar developments are currently underway in textual information processing.

The motivation of our work is the design of an automated system for an extraction task, namely the extraction of surface information in textual data. By surface information we denote information that does not require complex linguistic parsing in order to be extracted. For this, we use a combination of IR methods and stochastic language modelling. More precisely, our goal is to detect and extract sub-passages from on-line documents, and assign them *conceptual* labels representative of the information they carry. Labels are considered to be pre-determined in advance by the IE task definition, and we assume that there exists a labelled corpus from which to train our model (this set could be simply generated by asking the user to hand-label some documents to teach the model what the information extraction task is). Typical applications within this framework are intelligent routing of electronic mails or Usenet news postings, automatic summarization, highlighting, or text mining applications for the automatic recollection of data in the WWW.

Given the amount of data available in typical on-line collections, its diversity and its low informational content, we need to process data intelligently to locate the potentially relevant text passages; this can be accomplished with IR techniques. The work presented in this paper is original in that it uses IR to eliminate as early as possible irrelevant information. This is done in a hierarchical structure where IR modules reject irrelevant text successively at the document and paragraph levels. Feature selection is also carried out successively. At the top level, a stochastic model performs the surface IE task on the passages selected by IR modules.

In order to test our approach we use the MUC-6 corpus and associated Scenario Templates Task (ST) [5]. This

corpus contains articles from the Wall Street Journal. For each job appointment, reassignment or destitution appearing in these articles, there is an associated ST. An ST is an object which describes the event using a set of pre-defined fields (see Figure 1). The corpus contains 200 articles and 400 STs; 50% of articles and more than 80% of paragraphs are irrelevant to the extraction task (that is, they do not contain any of the text needed to fill the STs). In typical Internet applications the amount of irrelevant data is even greater.

To our knowledge there are no previous approaches combining hierarchical architectures for IR and stochastic language modelling for IE. Hierarchical architectures in IR have been used for example by Koller and Sahami [10], who implement two-level probabilistic decision trees for text categorization, and by Wiene, Pedersen and Weigend [21] who take a similar approach using neural networks. The combination of IE and IR techniques has been the subject of recent investigation in symbolic ML [17]. Our work on stochastic language models is inspired by that of Mittendorf and Shauble in IR [13], that of Imai et al. in topic-spotting [8] and that of Stévenin-Barbier and Gallinari on the use of NNs for surface IE in speech recognition [19].

In Section 2 we describe in more detail the IR component of our system; the two innovations here are the use of Neural Networks (Section 2.1) to compute the similarity measure and the implementation of a two-level feature extraction approach (Section 2.2). We discuss in Section 3 the use of stochastic models for the IE task and present some results.

2 Locating Relevant Text with IR

The extraction of information is computationally demanding, even for the simplest IE tasks and models. Loosely speaking, in order to extract information it must be «understood» to some degree, and this requires taking into account the inter-word dependencies in a passage, their order, etc. On the other hand, for a given information extraction task, it is often the case that most of the corpus text is irrelevant, and the searched information is confined in a few passages.

We hypothesize that locating relevant text is an easier task than extracting it, and we propose to use IR techniques as an intelligent pre-processing step to locate relevant information. More specifically, we deal here with the *routing* problem. Since «user queries» are defined in advance by the extraction task, no ranking of documents is needed but rather a binary choice of wether to throw away information or pass it to the IE module. We call a passage irrespectively a document or a paragraph (or other possible sub-sections of a corpus collection) when they are considered as a whole. Passages will be encoded into a single vector where word order and inter-word dependencies are lost. We will see in Section 2.2 that passage-vectors contain sufficient information for their discrimination.

For this particular application, a passage is considered relevant if there is at least one ST associated with it. Standard pre-processing techniques are used (a 320 words stop-list, tokenization including Porter-stemming, and frequency cut-off). The feature selection process is detailed in Section 2.2. Standard *tf-idf* is used to encode documents into vectors and Neural Networks are used to classify passages as described in Section 2.1.

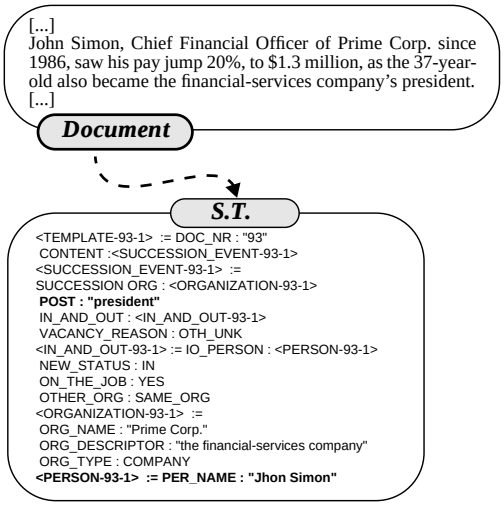

Figure 1: An IE example from the MUC-6 Scenario Template Task. The corpus consists of Wall Street Journal Articles (Documents) and their corresponding filled Scenario Templates (ST). We show in bold the only two fields (PERSON and POST) dealt with in this work.

2.1 Multi-Layer Perceptron Classification

We use Multi Layer Perceptrons (MLPs) [6] to classify passages. MLPs are one of the most general types of Neural Networks. They implement a large family of non-parametric functions and can fit a wide range of distributions in high-dimension. They can implement, within the same mathematical formalism, a large number of statistical classifiers (linear, logistic and nonlinear). Their parameters are determined automatically by the back-propagation implementation of the gradient descent optimization technique[1], which is an explicit error minimization technique. The use of MLPs has become widespread in several domains like pattern recognition, statistical modelling and signal processing and they have been used with success in large-sized routing problems [18]. One draw-back is the lengthy process of training, but this process needs to be carried out only in the development process, and requires little supervision. A more serious problem is that of model choice: the complexity of the MLP must be adapted to the difficulty of the task and to the amount of information present in the data; several techniques have been developed to deal with this problem.

With a single hidden layer and one output, a MLP takes the following form:

$$y(\mathbf{x}^s) = \sum_{j=1}^{H} \left[f\left(\sum_{i=1}^{T} x^s{}_i \cdot w_{ji} + \theta_j \right) \cdot w_{Oj} \right] + \theta_O \tag{1}$$

where $\mathbf{x}^s$ is the state-vector representation of document *s*, $x^s{}_i$ is its *i*-th component (i.e. the weight of the *i*-th term of document *s*), $y(\mathbf{x}^s)$ is the output of the network (which models the posterior probability of relevance of the document), H is the number of hidden units, w_{ji} and w_{Oj} are the input-to-hidden and hidden-to-output weights (there is one single output in our case), and θ_j and θ_O are the hidden and output unit bias. The function f is a non-linear squashing function, here the sigmoid function $f(x) = (1 - \exp(x))^{-1}$. If no hidden units are used, the model reduces to:

$$y(\mathbf{x}^s) = f\left(\sum_{i=1}^{T} x^s{}_i \cdot w_{Oj} + \theta_O \right) \tag{2}$$

which is equivalent to the logistic regression model (the difference being in the optimization algorithm used to find the weights; it has been repeatedly shown that MLPs outperform logistic regression in real-world applications [18]).

We use MLPs to obtain an approximation of the posterior probability of relevance of a passage $P(R|\mathbf{x})$. This measure can be used to rank passages in order of relevance and choose the most relevant for further analysis. Since rejected information is lost from further analysis, it is important that models reject as few relevant passages as possible. For this application we are interested in getting the highest precision for a predetermined -usually high- recall level. Note that this performance criterion is different from the training criterion which has been used for training MLPs.

2.2 Hierarchical Feature Selection

One of the main difficulties in IR is the selection of terms or features which will allow us to differentiate between relevant and non relevant passages. In order to improve the feature selection, we use the following hierarchical approach: features are first selected with respect to documents, documents are then classified, and the least relevant documents are eliminated from further analysis. Then feature extraction is carried out at the paragraph level using the remaining (mainly relevant) documents.

We use the U-measure [1, 2] for feature selection.This measure is similar to the well known X^2 measure, but does not take into account negative correlation. If we call p and p' respectively the number of relevant and irrelevant passages (in corpus *B*) in which the word w_i appears, q and q' the number of relevant and irrelevant passages in which it does not appear, and N the total number of passages (N = p + p' + q + q') then the u-value for word w_i, u_i, is:

$$u_i = \varphi_U(w_i, B) = \sqrt{N} \cdot \frac{(pq' - p'q)}{\sqrt{(p + p') \cdot (p + q) \cdot (q + q') \cdot (p' + q')}} \tag{3}$$

1. In our experiments we use a second order gradient descent algorithm called Scaled Conjugated Gradient [14], which offers the advantage of setting automatically the learning parameters.

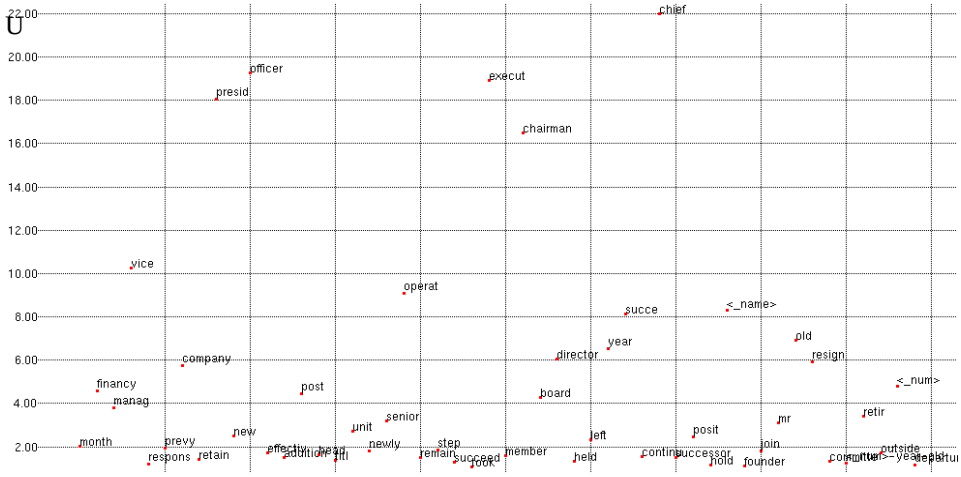

Figure 2: Paragraph Term U-values. Vertical axis indicates U-value, horizontal placement is random (for visualization)

We compute the U-value of all distinct words in the corpus and select as terms the 100 words with highest value[1]. This can of course be done at the document or paragraph level. In Figure 2 we represent the 100 terms selected at the paragraph level; we can observe that top ranking words are all semantically related with the extraction task (finding the name and position of each job appointment, reassignment or destitution appearing in the corpus). The standard IR technique of *tf-idf* weighting is then used to encode passages into 100 dimensional vectors $\mathbf{x}^j = (x^j_1, x^j_2, ..., x^j_{100})^T$ where j is the passage index and: $x^j_i = f(t_i, P_j, B) = tf(t_i, P_j) \cdot idf(t_i, B)^2$, where $tf(t_i, P_j)$ is the term frequency of term t_i in passage P_j and *idf* is the inverse document frequency of term t_i.

Because the U-measure and *tf-idf* encoding depend on the frequency of appearance of each term in relevant and non relevant passages, we expect the feature selection (and the resulting encoded vectors) to be more accurate if we first eliminate most irrelevant documents from the base. A hierarchical feature selection scheme should therefore lead to better classifier performances. We test this by classifying paragraphs with and without the document classification module. In the first case, documents are first filtered through the document classification module, and only those documents with high probability of relevance are considered for paragraph classification. In the second case, the document level is ignored; feature extraction is carried out only once, using all the paragraphs of the corpus. In Figure 3 (left) we show the precision-recall curves for these two systems: we see that the hierarchical system (*HS*) is clearly better than the non-hierarchical one (*NHS*) for regions of high values of recall, which are the regions in which we are interested. We are currently investigating the reasons of the degradation of performance in low recall regions.

In Figure 3 (right) we have represented terms with respect to their NHS (horizontal coordinates) and HS (vertical coordinates) U-values. In this representation, terms appearing on the very left (NHS=2) were eliminated in the NHS, and terms appearing to the very bottom (HS<2) were eliminated by the HS (they were present in the NHS). Some terms of interest are for example *company*, <*_num*> (any numeral), *search*, introduced by the HS system, or terms *newspaper, automotor, industry* eliminated by it. We note that all terms eliminated by it were low valued terms, but surprisingly some terms introduced by the HS ranked high (the terms *company* and <*_num*> for example).

The remaining terms were selected by both systems, but with differing relative importances (they constitute 84% of terms). Terms appearing exactly on the diagonal have the same importance in both representations, words to the right lost some relevance in the HS with respect to the NHS, and words appearing to the left gained relevance.

1. We have carried out tests with 50 and 200 terms also, but results in generalization were not as good. For this task, 100 terms seem to be adequate.

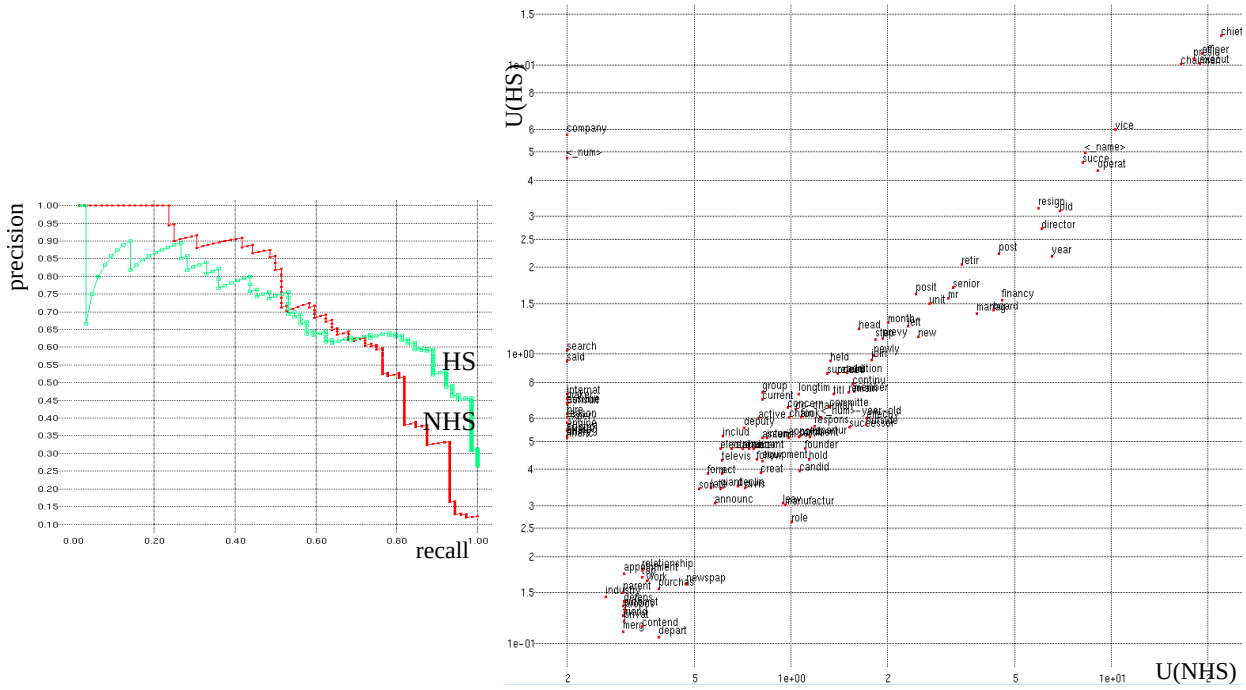

Figure 3: Hierarchial Feature Extraction. Precision-recall performance (left) and term U-values (right) for the hierarchical (HS) and non-hierarchical (NHS) models (see text for details).

3 The Stochastic Extraction Model

The static approach of IR (considering passages of text as vectors) is difficult to extend to IE. There has been some effort to apply IR techniques to IE tasks [12], but they are very limited and require very large amounts of data. The extraction of semantically meaningful information requires a more dynamic treatment of words, taking into account their context, relations, etc. Therefore, we need to consider at this level the modelling of word sequences. We describe here the use of Hidden Markov Models (HMMs) for this task. HMMs are well known discrete stochastic models, and have been applied with great success to a wide range of domains [3, 16]. They offer a convenient formalism to express context dependencies of sequential data, they can encode a large family of stochastic grammars, are computationally tractable and all of its parameters (except the structure of the model itself) can be learned from data.

3.1 Formal Description of HMMs

Formally, a (discrete symbol) HMM is characterized by the 5-tuple {N,M,A,B,π}, where:

- N is the number of states of the model, labelled {1, 2, ..., N},
- M is the number of distinct observations per state,
- q_t , w_t (used below) denote the state and output of the system at time t respectively,
- A={a_{ij}} is the state transition probability distribution matrix, and a_{ij} is the probability of going from state j to state i: $a_{ij} = P(q_{t+1} = j \,|\, q_t = i), \quad 1 \le i, j \le N$,
- B = {$b_j(k)$} is the observation symbol probability distribution, and $b_j(k)$ is the probability of producing symbol k (denoted as v_k) in state j : $b_j(k) = P(w_t = v_k \,|\, q_t = j), \quad 1 \le k \le M$,
- π = {π_i} is the initial state distribution, and π_i is the initial probability of state i: $\pi_i = P(q_1 = i), \quad 1 \le i \le N$.

Since we are dealing with text, *time* is interpreted as word position in a sequence. A passage of text is represented as the sequence of word-tokens **w** = (w_1 w_2 ... w_T), where T is the total length of the passage. The *distinct observations* are the terms (distinct words) in the corpus. States are left to be defined by the modeler. Word emission probabilities $b_j(k)$ and state transition probabilities a_{ij} are unknown initially but can be estimated with a corpus of text (possibly labelled with the states) using the Baum-Welch algorithm [3].

A HMM fully describes a probabilistic language model. Given a HMM model *M* and a state sequence $\mathbf{q}=(q_1 q_2 \ldots q_T)$, the probability of observing the sequence **w** is given by:

$$P_M(\mathbf{w}|\mathbf{q}) = \pi_{q_1} b_{q_1}(w_1)\left(\prod_{t=2,T} a_{q_{t-1}q_t} b_{q_t}(w_t)\right) \quad (4)$$

State sequences are usually not known (they are *hidden*). Given a HMM model *M* and an observation sequence **w**, we are usually interested in two things:

• the probability for the model to produce the observation:

$$P_M(\mathbf{w}) = \sum_{\mathbf{q}} P_M(\mathbf{w}|\mathbf{q}) \quad (5)$$

where the summation is over all allowed sequences of states **q**.

• the most probable sequence of states for the observation[1]:

$$^*_M(\mathbf{w}) = \arg\max_{\mathbf{q}} P_M(\mathbf{w}|\mathbf{q}) \quad (6)$$

Continuous Coding:

Learning probabilities directly on distinct words is difficult, because of their large number. This is a well recognized problem of stochastic language modelling [3]. We use the U-measure (3) to map the words of passages selected by the IR module into a one-dimensional space. With this measure the sequence **w** is mapped into the sequence of real values $\mathbf{u}=(u_1 u_2 \ldots u_T)$, using as a similarity measure the relative relevance of terms. With such continuous encoding of symbols, it is more natural to use continuous HMMs instead of the discrete HMMs described above. Conceptually, continuous HMMs are similar to their discrete counterpart, except for the symbol emission probabilities, $b_j(k)$, which are now defined as continuous functions over the symbol space. These continuous functions are usually defined as a gaussian mixture fitted on the observed probability distribution of the (continuous) symbol space [16]. We used this model in our HMM experiments.

3.2 Stochastic Extraction Model

If meaningful states are chosen, the HMMs may be used to extract information from text. Consider as an illustration the HMM model in Figure 4 (left) proposed by [13]. States are indicated by circles and possible state transitions are indicated by arrows. There are three states, the first and last representing the Irrelevant-Text concept (I) and the second representing the Relevant-Text concept (R). Given a paragraph $\mathbf{u}=(u_1 u_2 \ldots u_T)$ and using equation (6), we can find the most probable state-sequence $\mathbf{q}=(q_1 q_2 \ldots q_T)$; this sequence will be of the form (I...IR...RI...I), that is, a sub-passage of relevant text within two sub-passages of irrelevant text. In this manner each term u_i is mapped to the most probable concept q_i taking into account the entire sequence of observations and all possible state sequences. Furthermore, one may compute the probability of this sequence of words and states and use it as a similarity measure using equation (5). The obtained segmentation will be of course meaningful only if the HMM structure corresponds to a production model for the text analyzed.

As a first example we used this model to locate sub-passages regarding the *PERSON or POSITION* ST fields in the MUC-6 ST. Bellow the HMM, we show a portion of a relevant paragraph (Figure 4, middle left) with the phrases containing the relevant information in bold style. Following this text (Figure 4, bottom left) we see the most probable path (with words coinciding with the R state in bold style) of the HMM. For this example, the model has succeeded to extract the relevant sub-passage although is has continued to the end of the paragraph. Quantitative results are presented in the next section.

The previous example remains in the world of IR; even though we selected sub-passages, there was no *semantic tagging* of the information. In this case it is sought to *label* or classify the transcribed speech signal. This is precisely what we wish to do with respect to our IE task. The idea is to map the desired information fields to states in a more complex HMM, and use it to segment passages, assigning to each field the appropriate word sequences. As a more complex example of information extraction we introduce a model (Figure 4, top right) that labels portions of text with different concepts. Three concepts are used in this example: irrelevant (I), *PERSON* (R_1) *and POSITION* (R_2). The

1. It is not necessary to consider all possible segmentations to compute the probability of the most likely one (this computation would be of order of complexity N^T). The Viterbi algorithm [20] can be used to determine the most likely state sequence and its complexity is only of order N^2T.

last two correspond to fields of the MUC-6 ST (Figure 1). The *star* state assures that transitions happen only at the beginning of phrases[1]. While this is not necessary, it speeds up the search process and produces more coherent sequences. In the example (Figure 4 middle and bottom right), bold and underlined styles represent the *PERSON* and *POSITION* concepts respectively. The sequence of states obtained by the model is I-R1-R2-I, R1 coinciding with «Richard Furlaud» and R2 with «co-chairman (...) Bristol-Mayers Inc.» which are the desired *PERSON* and *POSITION* fields). Note that this model could be easily extended to incorporate other concepts by the addition of new states. However, as a preliminary step, we restricted our work to two relevant concepts.

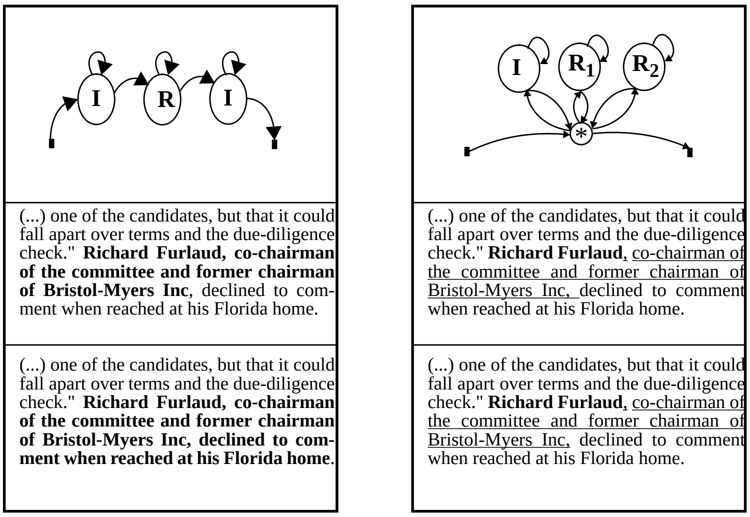

Figure 4: Two HMMs for language modelling (top) and an instance of the extraction task (middle and bottom). A section of a relevant paragraph and its desired tagging is shown in the middle and the obtained with the model is shown at the bottom. On the left, bold indicates relevant (R) text. On the right bold and underline indicate PERSON (R_1) and POSITION (R_2) states respectively.

3.3 Results

The evaluation of IR and IE systems is a difficult problem. As a first evaluation of our extraction module, we use the measures of precision and recall as follows. Let h be the number of words correctly labelled by the model, N the true total number of words of a given concept, and H the total number of words labelled by the model as being of that concept, then for this concept, precision = h/H and recall = h/N. In order to obtain different precision-recall points we fix different values for the state transition probabilities (increasing or decreasing the probability of staying in the concept state). In Figure 5 we present the precision-recall curves for the first (Figure 4 left) and second (Figure 4 right) models discussed. For the second model, we present two precision-recall curves: one for the POSITION field and one for the PERSON field. Evaluation is still quite preliminary, but it is encouraging. We see that simple models are capable of simple surface IE.

1. We define phrases loosely as sequences of words separated by punctuation marks. Labels where assigned to words of the relevant MUC-6 documents as follows: a paragraph containing strings matching at least two fields of an ST was considered relevant; phrases from relevant paragraphs containing at least one string matching a field of the associated ST where labelled with the field's name.

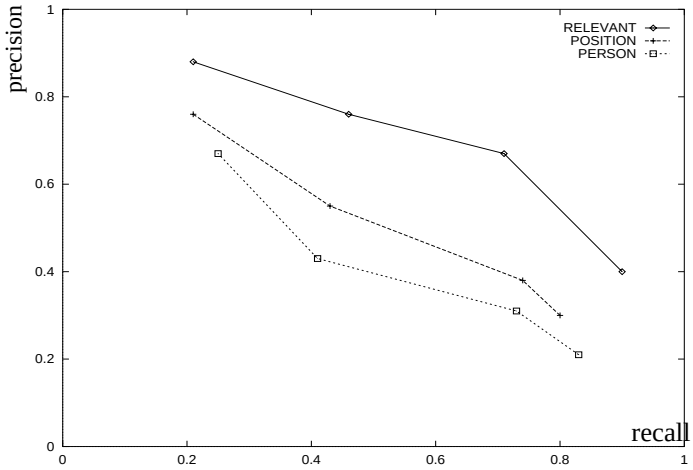

Figure 5: Preliminary Performances of the stochastic IE Module.

4 Conclusion and Perspectives

We have described a hierarchical approach to automatic information retrieval and extraction. IR techniques are used to eliminate documents which are irrelevant to the extraction task, then paragraphs. Finally, a stochastic model extracts (and labels) the sequences of words which contain the information to be extracted. We tested our models on a simplified version of the MUC-6 ST task. This approach could be used for example in surface analysis and routing of e-mail or Usenet posting and for text mining applications. Furthermore, our model skims, condenses and pre-codes textual data so that more complex conceptual taggers (such as those we developed for constrained information retrieval tasks [19]) can be re-used.

We are currently working on more sophisticated stochastic models to make a richer use of context. Parallel to this work, we are investigating approaches to learn simultaneously the different models of our hierarchy. In this context, we study coding schemes for adaptive feature selection. Rigorous evaluation of our models is underway.

References

[1] Andersen E. (1992) The Statistical Analysis of Categorical Data. Springer, Berlin.

[2] Ballerini J.P., Buchel M., Domeing R., Knaus D., Mateev., Mittendorf E., Schauble P., Sheridan P., and Wechsler M. (1996) SPIDER Retrieval System at TREC5. Proc. of the 5th Text Retrieval Conference TREC-5.

[3] Charniak E. *Statistical Language Learning*, a Bradford book, The MIT Press, 1993.

[4] Grishman R. (1996) Design of the MUC-6 Evaluation. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 13-33. Morgan Kauffman.

[5] Grishman R. et Sundheim B. (1996) Message Understanding Conference - 6 : A Brief History. COLING 96, 16th International Conference on Computational Linguistics. (Copenhagen 1996) vol. 1, 466-471.

[6] Hertz J., Krogh A. et Palmer R.G. (1991) Introduction to the Theory of Neural Computation. Addison-Wesley Publishing Company, Redwood City, CA.

[7] Hobbs J.R. (1993) The Generic Information Extraction System. In *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kauffman.

[8] Imai T., Schwartz R., Kubala F., Nguyen L. (1997) Improved Topic Discrimination of Broadcast News Using a Model of Multiple Simultaneous Topics. In ICASSP'97, 727-730.

[9] Knaus D., Mittendorf E., Schauble P. and Sheridan P. (1995) Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System. Proc. of the 4th Text Retrieval Conference TREC-4.

[10] Koller D., Sahami M.(1996) Hierarchically classifying documents using very few words.

[11] Lehnert, W. (1996) Information Extraction. (electronic publication: http ://www-nlp.cs.umass.edu/nlpgroup/nlpie.html)

[12] Lewis D.D. (1994). Data Extraction as Text Categorization: An Experiment with the MUC-3 Corpus. *In Proc. of the Third Message Understanding Conference (MUC-3)*

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

[14] Möller M.F. (1993) A scaled conjugate gradient algorithm for fastsupervised learning, *Neural Networks*, vol 6, 525-533.

[15] Pierracini R., Levin E. (1991) Stochastic Representation of Conceptual Structure in the ATIS Task. In 4th DARPA Workshop on Speech and Natural Language.

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

[17] Robertson M.A. (1997) On the Marriage of Information Retrieval and Information Extraction. *19th Annual BCS-IRSG Colloquium on IR Research* (Aberdeen 1997) 60-69 ,Furner J. Harper D. (eds.)

[18] Schutze, Hull (1995) A Comparaison of Classifiers and Dcoument Representations for the Routing Problem . In *TREC-4*, 1995.

[19] Stévenin-Barbier A. et Gallinari P. (1997) Semantic anticipation for understanding using neural networks, PACES / SPICIS (Singapore 1997).

[20] Viterbi A. (1967) Error bounds for convaolutional codes and an asymptotically optimum decoding algorithm. In *IEEE Trans. Information Theory*, IT-13:260-269.

[21] Wiener, Pedersen (1995). A Neural Network Approach to Topic Spotting. In *Proc. of the Foruth Anual Symp. on Doc. Analysis and Information Retrieval (SDAIR'95)*, 317-321.