Chapter 6

Markov Random Fields for Information Extraction

6.1. Introduction

In the automatic processing of languages, the ideal is to provide computers with the means to *understand* the texts, aiming to, little by little, leave it to the more modest and pragmatic objectives, which are formulated as *specific tasks*. The information extraction is typically one of these tasks. It aims to identify the factual information elements within a document, where the elements correspond to the fields that are previously defined from a form. In a way, it aims to fill the void that exists between the manner in which the humans apprehend the information, where the comprehension of natural languages plays a large part, and the manner in which computers process them, in other words in the form of typed data ordered in the structured files or in databases. In a review article on the subject, McCallum discusses an information *distillation* format [MCC 05].

To achieve such a task, several methods have been used. Since this is more and more the case for the set of linguistic engineering, these are the statistical models which are currently the most efficient. This is true only when we correctly reformulate the task such as an annotation or labeling problem. The best statistical models capable of learning data annotation are or CRF, *Conditional Random Fields* Markov random fields or conditional random fields (CRFs).

This chapter is an occasion to present the information extraction task and the statistical labeling models. The first two sections concentrate on the task, by discussing

---

Chapter written by Isabelle TELLIER and Marc TOMMASI.

its issues and the specific problems posed. The next four sections focus on the statistical models which give rise to definitions of Markov random fields. They constitute a theoretical and practical introduction on the annotation models in general, and the CRF in particular, which are increasingly used in many other contexts other than information extraction.

6.2. Information extraction

In this section, the problem of information extraction will be presented and the approaches attempting to solve it without calling upon machine learning techniques will be discussed.

6.2.1. *The task*

The information extraction, in terms of specific language task engineering, emerged at the end of the 1980s has become increasingly significant with the development of the Internet and numerical documents. Its aim is to automatically extract, from *text or semi-structured documents*, *factual information* that are ready to be ordered in the fields of a form.

At the original interest in this task, there is a series of MUCs (*Message Understanding Conferences*) which were developed between 1987 and 1998 under the direction of DARPA[1] (Defense Advanced Research Projects Agency). These conferences made available to the participants a text corpus as well as a list of fields to fill in. The participants had to supply a program that was capable of automatically filling the fields according to any element of the corpus. The programs were classified based on the function of their performance. Thus, for example (extracted from MUC-4 in 1992, translated in [POI 03]), from the dispatch of the press agency:

> *"San Salvador, 19 April 1989 (ACAN-EFE) – [text] The President of San Salvador Alfredo Cristani has condemned the original terrorist attempt on minister of justice Robert Garcia Alavo and has blamed the murder on the Farabundo Martí National Liberation Front. (. . .)"*

the following factual information will be extracted:

Date of the incident	19 April 1989
Location of the incident	El Salvador: San Salvador (City)
Author	Farabundo Martí National Liberation Front
Victim	Roberto Garcia Alavo

1 *Defense Advanced Research Projects Agency.*

The text of the agency has, therefore, been among the ones which have been initially targeted by the information extraction task. However, many other applications have since appeared, such as transforming small ads or specialized emails (e.g. announcing seminars or conferences) in standardized forms, automatically ordering in a database the coordinates of individuals that are known only by their Internet page or the *curriculum vitae*, and indexing scientific articles to constitute bibliographic databases (such as CiteSeer[2]) and so on [MCC 05].

These examples illustrate that information extraction has a very strong connection with recognition and the typing of named entities, and these words or groups are absent in usual dictionaries which identify either proper names (of people, places, and organizations, etc.) or measurable quantities (dates, numerical, or monetary values, etc.) which convey a large part of the informational content of certain texts, notably journalistic. The fields to fill in an information extraction task must be done by data of this nature. We recognize that the named entities in a text is not enough, and the role played by each of those in the related event to correctly fill the fields or the form must be found. Notably, this goes through the recognition of relations which maintains one another, and are often expressed by the predicates of the text. Spotting named entities is, therefore, not the only problem to resolve: the texts to process can also contain ellipses, pronominal references, or other types of anaphora that are often necessary to resolve to complete the task. All these problems will not be addressed in this chapter: extracting and typing of named entities, through the best actual methods from machine learning will be focused on.

In the 2000s, information extraction extended to new or growing types of documents such as: HTML and XML. These objects are often regrouped under the terms of *semi-structured documents*. These objects now have been specifically instructed to authorize several possible readers. In fact, they can be considered as character chains, and in this case fundamentally do not differ from the other texts, if it is not the presence of a supplementary vocabulary constituted of markers. These markers describe the tree structure. In this case, new notions such as *node* notions or even, for example, *next sibling node of the same father* become probable. The programs charged with extracting factual information of such documents are, in general, interested in accounting for this structure. It is, however, not as rigid and typical as those of databases, from where the *semi-structured* nomenclature comes from.

The term *wrapper* is sometimes used to designate an information extraction program which is founded on structure elements to identify certain information. It is, therefore, employed in the context of semi-structured documents, but not exclusively.

2 http://liinwww.ira.uka.de/bibliography/Misc/CiteSeer/

6.2.2. *Variants*

To characterize the different possible instances of an information retrieval problem, it is possible to classify them as input and output functions of the associated programs.

The input is specified at the same time by the representation of the considered documents, and by the valency of the values to extract. The documents can always return to a unity sequence, whether the latter are of a linguistic nature (word or word groups) or sequential nature (separators and markers). Semi-structured documents can otherwise be represented by a tree. The valency of the values to be extracted can be defined as the relationship between the units which constitute the documents and those waited for in the form. Do the fields that are to be filled in correspond to a unique unit, a unit portion, or a (connective) sequence of units? In the initial example, all the fields receive a sequence of words as a value. This choice is normally made for all "rough" texts, in other words unstructured texts. In this case, the *segmentation* of the documents seem to be a primordial preliminary phase, which is already assumed to be accomplished by the following. When the documents are represented by XML trees, the information to extract is found on the leaves. Even here, they can correspond to a leaf portion, an entire leaf of a sequence of "consecutive" leaves. A different level of granularity is found here, which resembles the same segmentation problem as before.

The output expected by an information extraction program can also take several formats. In the most simple case, also said to be unary, the data to be extracted are of a single and same type. Everything happens as if the form to fill in is composed of a single field. When it has n, nth-case is spoken of. But this is not all. The multiplicity of the expected responses must also be detailed, in other words the link between a document and the number of response instances it contains. If, for example, the problem is to recognize all the individuals, who are spotted by their name. Cited in a text or on a HTML page, this is therefore a multiple unary case (a single field with numerous instances). The example cited above is from a unique nth problem (several fields filled once by the document). To further complicate matters, certain situations are hybrid: the number of authors in a book can therefore be different from one book to another. . .

6.2.3. *Evaluation*

To evaluate the quality of an information extraction system, the measures used are the same as for the information extraction task, namely: the recall, the precision, and the F-measure. The specificities of information extraction, and its variations discussed in the previous section, make these measures much more problematic and are subjected to discussions furthermore. The human inter-annotation agreement is in fact often less stronger than in other contexts and the recognition of the "partially correct" or semantically equivalent values to the reference extractions are particularly delicate.

To have an idea of the debates to which these problems give rise to, and that can be detailed here, can be reported as an example in [SIT 04, LAV 04].

6.2.4. *Approaches not based on machine learning*

Before the machine learning techniques became more general, the information extraction task was processed by the *manual conception of specialized resources*. Typically, for the retrieval of named entities in texts, the necessary resources are ordered as follows:

– specialized dictionaries (list of proper names);

– retrieval patrons, in general, in the form of finite automated batteries or applied transducers *in cascade*, such as in the network of recursive transitions. In addition to spotting regular *patrons* such as PLC's, the transducers add annotations to texts, which provides useful indications to other transducers which are applied according to them.

This is, for example, how the Faustus system worked [HOB 97], one of the most famous participants of the MUC campaigns.

When extracting the information from HTML pages or XML documents, the patrons generally take the form of queries written in XQUERY or XSLT. The writing of these extractors are often complicated, difficult to maintain, and verify. To avoid writing grep, sed, and awk-based programs, diverse assisted conception software allows for the writing of patrons: Unitex[3] and Nooj[4] are therefore tools for multilingual linguistic engineering which integrate transducer editors and are often used to write extractors operating on rough texts, whereas XWrap[5] or Lixto[6] assist in writing wrappers for semi-structured documents. A platform for text and document management such as Gate[7] is also capable of integrating retrieval programs with diverse nature.

When they are carefully written, the patrons generally provide good result in terms or precision, and are quite poorer to recall. In fact, it is difficult to predict all the possible ways to express certain information. However, in all cases, the writing is a long and fastidious task, which requires technical and/or linguistic skills. The maintenance of patron-based systems is also problematic since there is an increase in their quantities. By applying the "manual" development strategy, a specialized extractor on a very restrained type of corpus is the best that can be expected, to be entirely reviewed as soon as the format, specialty language, or domain changes.

---

3 http://www-igm.univ-mlv.fr/ unitex

4 http://www.nooj4nlp.net/pages/nooj.html

5 http://www.cc.gatech.edu/projects/disl/XWRAP/

6 http://www.lixto.com/lixto_visual_developer/

7 http://gate.ac.uk/

6.3. Machine learning for information extraction

Over the course of the challenges on the development of applications, it has appeared more and more clearly that developing patrons in the hand to create information extraction was not a perennial solution. The more credible alternative comes from supervised machine learning. In this section, all the techniques coming from this approach which were envisaged will be discussed, before focusing on the reformulation of the information extraction problem which allows for the introduction of Markov random fields.

6.3.1. *Usage and limitations*

The usage of supervised machine learning was encouraged by the last MUC editions, which proposed the data issued from several different domains and languages, accompanied by the examples of retrieval results, without leaving sufficient time to the participants to enable them to manually develop the specific extractors. The majority of systems have participated in the latter of these conferences, in 1998, and thus already used, a stage or another of their development, a machine learning phase.

The benefit of the step is in the flexibility acquired: the same machine learning program can be applied to the data that has different (language, style, genre, structure, etc.) properties, and supplies each time an adapted extractor. In return, these programs require a consequent set of *labeled examples* in order to function, in other words input data (rough texts or semi-structured documents) associated with the corresponding filled forms. The collection and processing of these examples can also be lengthy and fastidious, but in general demands few technical skill than the direct writing of a extractor.

The diverse national (ESTER, DEFT, etc.) or international (TREC, CoNLL, ACE, etc.) challenges which have taken over MUC conferences all implicitly assume that the participants use techniques of this genre. They supply labeled data several weeks before publishing the real data on which the competition rests. The delay enables the process of automatic learning to take place, never writing directly by hand the extractor(s) making it possible to compete.

But machine learning is, however, not an easy cure for all to implement, and new problems arise. In fact, there are few applicable supervised machine learning algorithms: the latter accomplish for the most part of *classification* or *annotation* tasks. The information extraction task must often be reformulated in a manner, so that it is brought back to a series of instances of these numerous generic problems. Reformulation examples will be seen in the later stages of this chapter. Once the first step is accomplished, other difficulties appear. Therefore, the available learning data are only *positive examples*, in other words correctly extracted data. The majority of machine learning programs also have the need for *negative examples*. In certain cases,

depending on the variants (see section 6.2.2), the supplied data enable the introduction of negative examples. Thus, when the multiplicity of the expected response is unique then all responses other than the correct one is a negative example. However, in this case care must be taken not to introduce an imbalance between the classes which will be detrimental to the functioning of the algorithm.

Finally, as always when considering the intervention of automatic learning in a language engineering task, the problem of accounting for the knowledge or external linguistic resources arises. What do the specialized dictionaries or patiently harvested patrons do during manual processing? How to reinvest them in a program which contents itself with learning from example? It will be seen that the CRF suggest certain interesting paths in order to respond to these questions.

6.3.2. *Some methods*

Numerous supervised machine learning methods have been employed to address the information extraction task [SAR 08]. Certain methods are adapted only to a certain type of input data (text or semi-structures), whereas the others can be applied to any. Here, only certain methods will be discussed, before developing those which rely on conditional Markovian fields.

First of all, since the majority of the extractors written by hand take the form of the patrons, it is natural to try and infer directly such patrons according to available examples. Historically, these are first and foremost symbolic machine learning strategies inspired by these findings which have been attempted: the pioneering system of Rapier can be cited, issued from inductive logical programming [CAL 03]. Grammatical inference, which is linked with the machine learning of language representation models (such as regular, automated expressions, and grammar from diverse classes) is another possible symbolic approach. When applied to the texts, it gave few convincing results (see however, [FRE 97, FRE 00a]). However, it showed itself to be efficient for learning tree wrappers [KOS 06, CAR 07]. Patrons can also be introduced from the analysis and the generation of sequences, using *suffix tree* algorithms [RAV 02] or using methods inspired by searching for sequential data [CHA 09].

However the most common approach, as it is applicable to all the types of input data, consist of reformulating the problem as probable to *supervised classification*, for which numerous systems and algorithms (neuron network, decision trees, Bayesian classifiers, SVM, etc.) already exist. For the texts, the aim is to rank the linguistic units of which the constituent is “data to extract” or not. It has been seen that retrieval data often corresponds to a *sequence of units*. It, therefore, seems to be more efficient to rank the *separators* between units, as long as such that they correspond to the “start” of a piece of data to extract, to the “end” of such data or if they are “neutral”, in other words situated in the false-relevant for the extraction. It is this approach which is used

in BWI [FRE 00b]. It is also possible to try and rank the *separator couples* such that they do or do not framework a text unity sequence to extract, as in the Elie system [FIN 04]. These strategies have been generalized for information extraction in trees in [MAR 07].

6.3.3. *Annotate to extract*

For the remaining of this chapter, the approach involves focusing on another different reformulation of the information extraction task, which consists of returning to an *annotation* problem. With such a problem, the initial data breaks down into distinct units, and each of those must be associated with a *label* belonging to a new finite vocabulary. In a sense, the approach discussed at the end of the previous section is a particular annotation case, in which the units considered are the separators. In this section, in the case if the units are of a linguistic nature (words or groups of words), but they can also be of a structural nature (separator, label). The fundamental difference between this task and that of classification is that, in the context of an annotation, it is probable to take into account the *relationships between labels* to assign them correctly. An annotation is, therefore, not a series of independent classifications: it is the set of all the labels associated with the units which is the aim of learning, and not each of them independently. This is what enables the statistical process which will be developed further on.

Numerous linguistic engineering tasks enter the framework: this is of course in the case of morpho-syntactic labeling, where it is the association of each linguistic unit of a text to a category among "common names", "verb", "adjective", and so on with supplementary morphological information as the genre, number, conjugation, and so on. It is in fact evident that, in this case, the labeling to associate with a particular word strongly depends on the labels associated with the neighboring words in the same sentence.

To bring back an information extraction task to an annotation task, a set of labels must be associated with each field to extract, in general chosen among the letters BCEO (B for *Begin*, C for *Continue*, E for *End*, and O for *Other*) or BIO (I for *Inside*) [SAR 08]. A simple label couple IO (*In*/*Out*) does not enable the distinction to be done between a sequence of units which constitutes a single value for a field to fill (e.g. a unique entity such as "19 April 1989" or "Olympic Games") and a sequence of distinct values for the same field (e.g. an enumeration of distinct proper names).

In the following example, which describes an event, the label starting with P is relative to a location (*Place*) and those starting with E to an event relating to the nature of the event:

London	will	host	the	next	Olympic	Games
PB	O	O	O	EB	EI	

From such labeling, it is easy to fill a form on the described event. The information extraction from the semi-structured documents can, in general, also be described as an annotation task, such that the units coincide with the tree nodes to label at this time.

6.4. Introduction to conditional random fields

Conditional Random Fields constitute a family of statistical models which enable the data label learning[8]. They have common traits with numerous formalisms defining joint or conditional distributions. They therefore lend themselves to models based on logistic regression and on the principle of maximum entropy [BER 96], often used when automatically processing languages, the definition of a set of parameters associated with *characteristic functions.* With the Bayesian or hidden Markov models, they share the data of a structuring graph and a good number of algorithmic solutions to inference problems. For these reasons, Markov random fields can be introduced at the same time looking at it from the maximum entropy angle and that of the hidden Markov model angle. In this section, the entropy maximum models will be focused on, but the link with the hidden Markov models will also be established in section 6.6.

6.4.1. *Formalization*

For introduction, the previous example of the Olympic games in London will be re-addressed. The modeling of the problem consists of considering a sentence of six words and implementing $\boldsymbol{x} = (x_1, \ldots, x_6)$ of a field on six random variables $\boldsymbol{X} = (X_1, \ldots, X_6)$. These X_i variables are said to be *observation* variables. The labeling is the implementation of $\boldsymbol{y} = (y_1, \ldots, y_6)$ of random variables $\boldsymbol{Y} = (Y_1, \ldots, Y_6)$:

$\boldsymbol{x}$:	London	wil	host	the	next	Olympic	Games
$\boldsymbol{y}$:	PB	O	O	O	EB	EI	

Once the $\boldsymbol{x}$ variables are observed, the probabilistic approach of the problem goes through the definition of the conditional probability:

$$\mathrm{P}(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) \qquad [6.1]$$

It is usually assumed that P belongs to a given distribution class, often parametric. This class conditions the resolution algorithms of two main problems to resolve: on the

8 Annotation or labeling is interchangeably used to designate the operation which consists of associating a label to data.

one hand, the determination of the labels to associate with a new observed sequence of words, and on the other hand, the identification of P from the labeled examples (in other words, the training of the model for the identification of its parameters). The difficulty of these problems also depend on the class to which P belongs.

PROBLEM **Pr1**.– Inference. *Given data P and $\boldsymbol{x}$, finding realization $\boldsymbol{y}$ which maximizes the conditional probability* $\mathrm{P}(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x})$.

PROBLEM **Pr2**.– Learning or training. *Given a finite example of S "associated annotation observation" couples with the form* $\{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$ *with unknown P, to identify the P parameters.*

6.4.2. *Maximum entropy models*

Taking into account the knowledge of the domain in the statistical model is often translated by the definition of characteristic functions taken from a finite set. These are most of the time functions with values of $\{0, 1\}$ such as:

$$f^i(y_i, x) = \begin{cases} 1 \text{ if } x \text{ with a capital letter and } y \neq \text{“O”} \\ 0 \text{ if} \end{cases}$$

This function applies to a position i in a sequence, for all the data pairs x and y. In this example of section 6.4.1, it gives a result 1 for positions $i = 1$, $i = 5$, and $i = 6$, and is 0 everywhere else. It is also 0 in the first position of a sentence which does not begin with a named entity. The application of such a characteristic function to a sequence consists of in fact performing such calls to each position in this sequence and combining the results. In certain cases, the useful knowledge for labeling expresses itself through conditions on different positions (but also neighboring) of the word which the label will predict. The characteristic functions are therefore a little more complex:

$$f^i(y, \boldsymbol{x}) = \begin{cases} 1 \text{ if } x_i = \text{“Games”} \text{ and } x_{i+1} = \text{“Olympic”} \text{ and } y = \text{“EB”} \\ 0 \text{ if not} \end{cases} \quad [6.2]$$

The definition of the statistical model is therefore fundamentally based on the maximum entropy principle: it is the $\hat{P}$ distribution which contains less information, and imposes few assumptions on the data (therefore, here, limiting itself to the knowledge carried by the characteristic functions), which must be chosen:

$$\hat{P}(\boldsymbol{Y} \mid \boldsymbol{X}) = \underset{P}{\operatorname{argmax}}\, H_P(\boldsymbol{Y} \mid \boldsymbol{X}) = \underset{P}{\operatorname{argmax}} - \mathrm{P}(\boldsymbol{Y}|\boldsymbol{X}) \log \mathrm{P}(\boldsymbol{Y}|\boldsymbol{X}) \quad [6.3]$$

With this assumption, the **Pr2** problem can be reformulated as an optimization problem under the limitation that $\hat{P}$ is a distribution[9]. Each characteristic introduces a constraint in the optimization problem. The expression of its solution therefore goes through the Lagrange multiplier introduction θ_k^i, which represents the coefficients associated with characteristic functions f_k^i. The definition of the model, therefore parametered by its θ_k^i, for each Y_i results in a characteristic [KLI 07, BER 96] format of the maximum entropy models:

$$\mathrm{P}(Y_i = y_i | \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z(x)} \exp\left(\sum_k \theta_k^i f_k^i(y_i, \boldsymbol{x})\right) \quad [6.4]$$

where $Z(x) = \sum_{y \in \mathcal{Y}} \exp\left(\sum_k \theta_k^i f_k^i(y, \boldsymbol{x})\right)$ is only a normalization coefficient. Instinctively, each θ_k^i coefficient characterizes the "importance" of the associated f_k^i function. A similar form is found for the Markov random fields.

The problem of **Pr1** can therefore be easily resolved. Furthermore, it is unnecessary to calculate $Z(x)$ to find the best y_i and the algorithmic difficulty of this approach stays simple: for each Y_i variable and for each possible value of y_i in its $\mathcal{Y}_i$ domain, in fact the $\sum_k \theta_k^i f_k^i(y_i, \boldsymbol{x})$ must be calculated. Let us consider that each value of f_k function can be calculated in a constant time[10], the algorithm is linear in relation to the number of variables, number of characteristic functions, and the cardinality of $\mathcal{Y}_i$.

In this case, the **Pr2** problem can also be resolved without too many difficulties. A sample of supplied data labeled by S, finds the best parameters, in other words those which maximize the entropy, return to search for the solution to the corresponding likelihood maximization problem [NIG 99, PIE 97]. The model described in equation [6.4] is an exponential product weighted by the parameters θ_k, the likelihood is convex and it is therefore possible to find this unique maximum, for example by a descending gradient algorithm.

Linking the parameters. The model proposed till now relies on a field of six random variables, each associated with sentence labels of six observed words. How to predict in this case the labels with words of seven words or more? In this equation [6.4], each Y_i variable of $\boldsymbol{Y}$ has its own set of characteristic functions and parameters. A solution consists of imposing a set of characteristic functions *common to all* $\boldsymbol{Y}$ *variables, no matter its position*. The functions are no longer indexed by the position, but the position becomes one of arguments and the θ parameters are assumed to be the same for all the variables:

9 Each value is between 0 and 1 and their sum is 1.

10 This is an assumption kept for the remaining of the entire chapter.

$$P(Y_i = y_i|\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z(x)} \exp\left(\sum_k \theta_k f_k(y_i, \boldsymbol{x}, i)\right) \quad [6.5]$$

Here a finite $f_1, \ldots, f_K$ set of functions is considered, identically applying for all Y_i variables.

6.4.3. *Hidden Markov models*

A critical importance is often advanced at the encounter of the approach by the maximum entropy model: it does not take into account the data structure, of their sequential organization for example. The dependencies between the successive labels are therefore not currently integrated in the model. In this sense, it is more a *classification* model than an *annotation* model. The hidden Markov models[11] constituent of a second approach, very classical, for processing the problem of sequence labeling, which begins to correct this default.

In a hidden Markov model, it is the joint probability $P(\boldsymbol{X}, \boldsymbol{Y})$ which is represented. Using the definition of the conditional probability and with the help of the decomposition rule, the conditional probability of a sequence of six Y_i variables is written:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{P(\boldsymbol{X}, \boldsymbol{Y})}{P(\boldsymbol{X})} \quad [6.6]$$

$$= \frac{1}{P(\boldsymbol{X})} P(Y_1) P(X_1|Y_1)$$

$$\prod_{i=2}^{6} P(X_i|Y_i, X_{i-1}, Y_{i-1} \ldots, X_1, Y_1) P(Y_i|X_{i-1}, Y_{i-1} \ldots, X_1, Y_1) \quad [6.7]$$

As this formula shows, researching the $\hat{\boldsymbol{y}}$ value which maximizes $P(\boldsymbol{Y}|\boldsymbol{X})$ very fastly becomes impractical with the length of the sequence and cardinality of $\mathcal{Y}$. In the hidden Markov models, a set of conditional independence assumptions is added to limit this difficulty. It is assumed that (i) all X_i is only dependent on Y_i and that (ii) all Y_i is conditionally independent of all other Y_j to Y_{i-1}. A simplified expression is therefore obtained by:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{P(\boldsymbol{X})} P(Y_1) P(X_1|Y_1) \prod_{i=2}^{6} P(X_i|Y_i) P(Y_i|Y_{i-1}) \quad [6.8]$$

In this case, the parameterization thus consists of determining the set of $P(X_i|Y_i)$ and $P(Y_i|Y_{i-1})$ probabilities and the input probability $P(Y_1)$.

11 See their definition in Appendix.

The addition of these assumptions allows for the solutions of the two **Pr1** and **Pr2** problems in polynomial terms, provided it is correct: the efficient algorithm, known as dynamic programming, is based on the factorization of equation [6.8] (for more details, refer to Appendix which presents the hidden Markov models). This technique is found in the case of Markov random fields.

The hidden Markov models are no longer exempt from criticism. The first door on the assumptions is associated with the $\boldsymbol{X}$ variables which are finally observed. The model forbids the expression of *dependencies between the words*, although their implementation is observable. It is therefore impossible to condition the labeling of a word in the presence of other words in the sentence, whereas the latter are known. The criticism signals that the hidden Markov models implement a more difficult task than those initially targeted: they produce a *data generation* model (using the calculation of $P(\boldsymbol{X}, \boldsymbol{Y})$) and by consequence the modeling of observed data too, whereas the annotation task does not require it. It will be seen that the Markov random fields are a way to respond to these two criticisms.

The hidden Markov models, the Markov random fields, and even the maximum entropy model presented here come under the class of *graphical models*, qualified as such not because it can be given a graphical representation, but to signify that the relations between their underlying variables form a graph. They will be quickly discussed in the following section.

6.4.4. *Graphical models*

In a graphical model, the random variables of the annotation problem become the nodes of a graph, and the absence of edges between the two nodes translates to a conditional independence relationship between the corresponding variables, given the other graph variables. Without a particular assumption, all the variables are dependent on one another and the graph is complete: for all node pairs, there is an edge that links them. Let us consider things from the hidden Markov model perspective (section 6.4.3), the graph obtained is that of Figure 6.1. It is a directed graph which translates the *generative* nature of the model: the label (Y) in fact "generate" the words (X).

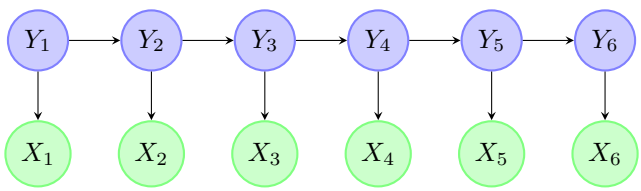

Figure 6.1. *Graph of the Markov chain models*

The relationships underlying the conditional independence models are in fact used to rewrite the equation [6.6]. In the case of hidden Markov models, it has been seen that they are rewritten in the equation [6.8]: the probability $P(\boldsymbol{X}, \boldsymbol{Y})$ is therefore written as a product of conditional probabilities of which each of the factors translates the dependence of a variable in relation to its antecedent (or father) in the graph.

Under the assumption of maximum entropy models (section 6.4.2), the graph is even more simple as all the Y variables are conditionally independent of all the $\boldsymbol{X}$ variables, which gives the graph of Figure 6.2. For reasons of commodity for the representation, the $\boldsymbol{X}$ node represents a click (in other words, a sub-graph completely connected) of all the $\boldsymbol{X}$ variables, and the edge between a Y_i and $\boldsymbol{X}$ constructs a complete click between all these nodes. The graph, this time, is not directed since the dependencies in this case are expressed by functions in which y_i and all the $\boldsymbol{x}$ elements play a symmetrical role, for each i value (see for an example, equation [6.2]).

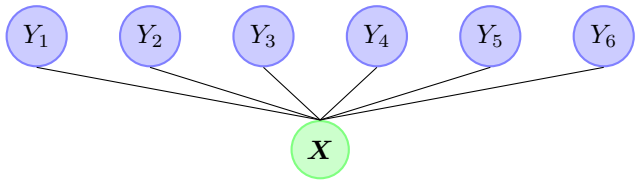

Figure 6.2. *Graph of the maximum entropy model. Here, the* $\boldsymbol{X}$ *node represents a click that contains all the variables of* $\boldsymbol{X}$

In the case of non-directed graphs, the probability of the field of variables is not written in the form of a product of probabilities, as is the case in the directed models, but in the form of a product of positive functions arbitrary on the $\mathcal{C}$ set of the clicks of the graph:

$$P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \Psi_c(\boldsymbol{y}_c, \boldsymbol{x}_c) \qquad [6.9]$$

Here, Z is always a normalization factor and $\boldsymbol{x}_c$ and $\boldsymbol{y}_c$ designate the implementations of the variables fields $\boldsymbol{X}_c$ and $\boldsymbol{Y}_c$ of band c. Each Ψ_c is called as the potential function associated with band c. For this maximum entropy model, the given expressions can be assimilated by equations [6.4] and [6.9]. The first is written for a single Y_i variable. Let us note that Y_i are in different bands: Y_i being independent of $\boldsymbol{X}$, there is a probability $P(\boldsymbol{Y}|\boldsymbol{X})$ written as a product according to the graph bands. Furthermore, in equation [6.9], if the $\boldsymbol{X}$ values are observed then the normalization factor Z depends on these values when evaluating the conditional probability $P(\boldsymbol{Y}|\boldsymbol{X})$. Finally, noting that the potential functions are only specific to the maximum entropy model, where they are written in the form of an exponential of a linear combination of characteristic functions.

Putting these two formalisms, and soon those of the Markov random fields, in the framework of graphical models enables the presentation of a uniform view of the algorithms which respond to the **Pr1** and **Pr2** problems. When calculating the marginal probabilities such as $\mathrm{P}(\boldsymbol{Y}_1 = y_1|\boldsymbol{X} = x) = \sum_{y_2,\ldots,y_6} \mathrm{P}(\boldsymbol{Y} = y|\boldsymbol{X} = x)$, the conditional independent relationships allow for the sum to be rewritten, resulting in an efficient calculation. For example, in the case of hidden Markov models, the sums "pushed" as far as possible in the product, give an expression with the following format:

$$\mathrm{P}(\boldsymbol{Y}_1 = y_1|\boldsymbol{X} = x) = \frac{1}{\mathrm{P}(\boldsymbol{X})}\,\mathrm{P}(Y_1)\,\mathrm{P}(X_1|Y_1)\left(\sum_{y_2}\mathrm{P}(X_2|Y_2)\right.$$
$$\left.\mathrm{P}(Y_2|Y_1)\left(\sum_{y_3}\mathrm{P}(X_3|Y_3)\,\mathrm{P}(Y_3|Y_2)\ldots\left(\sum_{y_6}\mathrm{P}(X_6|Y_6)\,\mathrm{P}(Y_6|Y_5)\right)\right)\right)$$

This rewriting makes an optimization of the calculation through dynamic programming seem to be possible. In fact, the most embedded in the loops identified by the sums, namely $\sum_{y_6}\mathrm{P}(X_6|Y_6)\,\mathrm{P}(Y_6|Y_5)$, can be memorized as it does not depend on other intermediate index loop variables smaller than five. By repeating this memorization to each intermediate factor, and such that constant access to the memorized factors is possible, and the difficulty falls to a second degree polynomial.

It is the underlying graph of the probabilistic definitions which make this factorization possible. This property is essential for efficiently calculating marginal probabilities. We note that the definition of the conditional independence graph with the $\boldsymbol{X}$ variables is not beneficial when representing $\mathrm{P}(\boldsymbol{Y}|\boldsymbol{X})$, since the achievements of $\boldsymbol{X}$ are observed. In the following, only graphs on the output variable $\boldsymbol{Y}$ will be considered. This signifying that all the variable of the $\boldsymbol{X}$ are implicitly linked to all those of the $\boldsymbol{Y}$ field.

6.5. Conditional random fields

6.5.1. *Definition*

There are two sets or fields of random variables: $\boldsymbol{X}$, representing the inputs or observables and $\boldsymbol{Y}$, representing the outputs or annotations. Where G, a non-directed graph of the sets of nodes is the set of random variables $\boldsymbol{Y}$ and where $\mathcal{C}$ is the set of bands of graph G, the variable fields of a c band are written as $\boldsymbol{X}_c$ and $\boldsymbol{Y}_c$ and their outputs are $\boldsymbol{x}_c$ and $\boldsymbol{y}_c$. The Markov random fields are representative models of a conditional distribution class $\mathrm{P}(\boldsymbol{Y}|\boldsymbol{X})$, which factorizes itself according to G:

$$\mathrm{P}(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{c\in\mathcal{C}}\Psi_c(\boldsymbol{y}_c, \boldsymbol{x}) \qquad [6.10]$$

$$\text{with } Z(\boldsymbol{x}) = \sum_{\boldsymbol{y},\, y_i \in \mathcal{Y}} \prod_{c \in \mathcal{C}} \Psi_c(\boldsymbol{y}_c, \boldsymbol{x}) \qquad [6.11]$$

In the Markov random fields, as in the maximum entropy models, the knowledge is translated by the characteristic functions and the maximum entropy principle is applied. In addition, the parameters are linked so that each potential function Ψ_c is written as:

$$\Psi_c(\boldsymbol{y}_c, \boldsymbol{x}) = \exp\left(\sum_k \theta_k f_k(\boldsymbol{y}_c, \boldsymbol{x}, c)\right) \qquad [6.12]$$

The parameters are $\Theta = (\theta_k)_k$. As already done for equation [6.5], linking the parameters leads to defining characteristic functions with the band (or an identifier of this) in question[12]. In this manner, a particular band of parameters θ_k and characteristic functions f_k is dissociated which apply. As a consequence, the inference and training of a Markov random field are not limited to the data (therefore, fields of variables) with identical size and structure.

The Markov random fields bring solutions to the problems posed by the models presented in the previous sections. In fact:

– they are adapted to the *structured data*, composed of items which maintain the relations with one another, as for example, the words of a sentence or the nodes of a tree. By applying itself on the field of variables, this allows for the overall processing of the annotation problem of each element of a given structure: for this it suffices that an output variable Y_i is associated with the label of each element. The data of a graph enables the expression of non-trivial conditional independence relationships between these labels. With this, it overtakes the maximum entropy models in section 6.4.2 which realize independent classifications and not a real overall annotation;

– the Markov random fields directly model a conditional probability. Information on the manner of which the data labeling realized is enough and it is not necessary to describe how these data are generated. This removes, in particular, the strong limitations of the hidden Markov models: the (observed) *labeling of an element of the data can arbitrarily totally or partly depend on the data*, all the while having practicable difficulty (on dependence reserves limited on the outputs).

In the following section, it will be studied how and under which limitations it is possible to efficiently resolve the two inference and training problems (see page 194) for the Markov random fields. All the difficulty of the inference problem is already found in the $Z(\boldsymbol{x})$ calculation. The summation on all the possible values of $\boldsymbol{y}$ in fact trains a combination explosion: if n is the size (in variables numbers) of field $\boldsymbol{Y}$, then

12 In [SUT 06], the authors speak of *features templates*.

there are different possible $|\mathcal{Y}|^n$ labelings. How this calculation can nonetheless be effectively realized, will be studied.

6.5.2. *Factorization and graphical models*

Let us consider a graph G which is represented in Figure 6.3, to the left. Only the node Y_4 assures the connection between the two "parts" of G. If this node is taken away, then the G separates into two disjointed sub-graphs: $G_{1,2,3}$ and $G_{5,6}$. This separability translates a conditional independence of all the variables of $G_{1,2,3}$ in relation to the variables of $G_{5,6}$ (and vice versa)[13]. The separability can be expressed in relation to a single node, as in this example, or even a set of nodes, respectively. In this example, there are four maximal bands[14]: the two sub-graphs $G_{1,2,3}$ and $G_{5,6}$ and the two bands with two elements $G_{3,4} = \{Y_3, Y_4\}$ and $G_{4,5} = \{Y_4, Y_5\}$. The coefficient of the Z normalization can be written, as done in the case of hidden Markov models, by "pushing" the sums as far as possible under the products:

$$\sum_{\boldsymbol{y},\, y_i \in \mathcal{Y}} \prod_{c \in \mathcal{C}} \Psi_c(\boldsymbol{y}_c, \boldsymbol{x}) = \sum_{(y_1, y_2, y_3) \in \mathcal{Y}^3} \Psi_{G_{1,2,3}}(y_1, y_2, y_3, \boldsymbol{x}) \left(\sum_{y_4 \in \mathcal{Y}} \Psi_{G_{3,4}}(y_3, y_4, \boldsymbol{x}) \left(\sum_{y_5 \in \mathcal{Y}} \Psi_{G_{4,5}}(y_4, y_5, \boldsymbol{x}) \left(\sum_{y_6 \in \mathcal{Y}} \Psi_{G_{5,6}}(y_5, y_6, \boldsymbol{x}) \right) \right) \right)$$

This factorization also rests on the order of the following variables in which the sums are pushed on the product: this is known as *variables elimination order*[15]. In the previous calculation, the Y_1 to Y_6 order has been followed. Starting with Y_4 for example, would not enable the maximum exploitation of the evidenced separation. Owing to this factorization, it goes from a $|\mathcal{Y}|^6$ to a $|\mathcal{Y}|^3$ difficulty, where 3 is the number of the largest G band[16] (and corresponds to the enumeration of all the triplet values possible for the Y_1, Y_2, and Y_3 variables).

---

13 The Hammersley–Clifford theorem [HAM 71] establishes the fact that separability in the graph, factorization according to the band of the graph and conditional independence are three equivalent properties which enable the definition of the same class of distributions.

14 The bands included in these maximal bands do not intervene in the factorization which follows and their contribution to the probability can be easily integrated in that of the maximal bands.

15 This approach is known for the case of graphical models or Bayesian networks under the name of variables elimination algorithm [SHA 90].

16 The general principle of the algorithm is not given here. It is described on page 199 in the case of hidden Markov and consists of conserving the calculations already done.

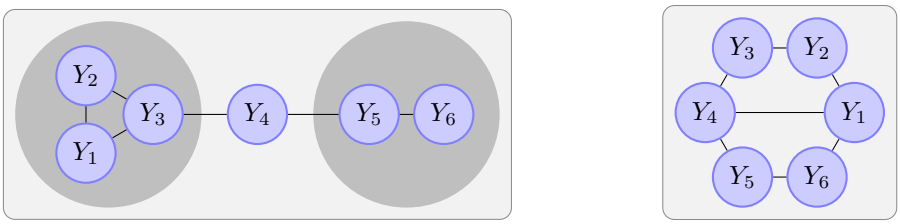

Figure 6.3. *Conditional and separability independences*

In the case of more complex graphs, as well as for the calculation of any marginal probability, the factorization and choice of the order of variables elimination seem to be less readily. For example, for that of Figure 6.3(to the right), it is necessary to regroup at least two nodes to be able to cut the graph into two disjointed sub-graphs. According to the choice of these two nodes, it is possible to obtain different factorizations, which has given rise to different complexities for the calculation of Z.

A uniform approach to evaluate Z or all marginal probability relies on the construction of a *junction tree* and on the application of the *message passing* dynamic programming algorithm, presented in the following sections. The conditional independence graphs with the form of a chain or tree have sufficient properties to enable the direct application of this dynamic programming algorithm. The message passing general algorithm is a generalization of those which apply to the hidden Markov model (*forward-backward*) or the PCFG[17] (*inside-outside*, α-β).

6.5.3. *Junction tree*

A junction tree J for a graph G is a graph of which the nodes are band unions obtained from G and which satisfy the following conditions:

– it is a tree, therefore there always exists only a single path between any two J nodes;

– all bands of G is contained in the J nodes;

– for all node couple A and B of J, if a Y_i variable belongs to A and to B at the same time, then it also belongs to all J nodes that are found on the A and B path.

The junction tree of a given graph is not unique. The tree reduced to a single node including all the bands of the graph is therefore always a junction tree, even if it is not very useful for factorizing the calculations. Figure 6.4 gives two tree junction examples for the same graph.

17 Probabilistic context-free grammars, see for example [MAN 99].

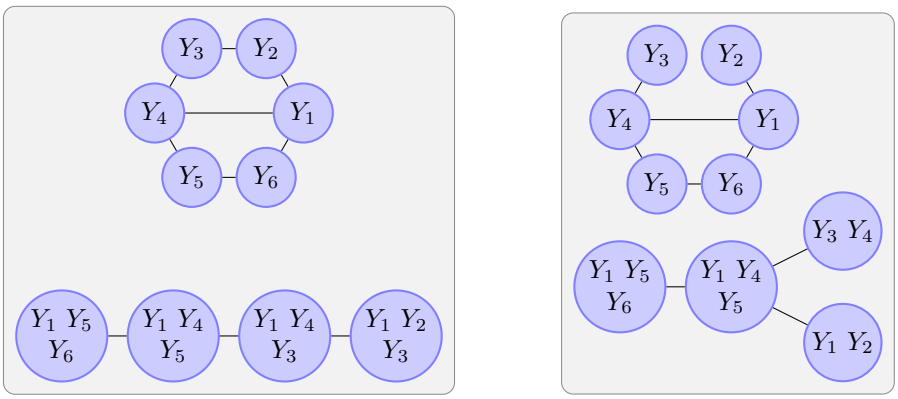

Figure 6.4. *Examples of two graphs (above) and junction trees (below)*

The construction algorithm of a junction tree of a graph, quite classical in graph theory, is presented as an example in [COW 99]. It relies on the determination of the order of elimination of variables which enables the triangulation[18] of graph G. With the bands thus created, a graph of weighted bands in which a recovery tree is selected maximize the weighting. The best tree is the one where the largest band has the smallest cardinality, but the problem of the determination of this tree is NP-dur[19]/ Luckily, for certain graph classes, it is possible to construct an optimal junction tree efficiently.

In a junction tree, by construction, to each j node there corresponds a set of S_j bands formed in the original graph following its triangulation j is therefore associated with a new potential function Ψ'_j which represents the product of all the S_j potential functions. Through this association, several original Ψ factors are repeated in different Ψ'. The Ψ' definition must therefore be adapted in order to avoid this duplication of factors. For example, for the graph of Figure 6.4(to the right), there are the following potential functions $\Psi_{3,4}(y_3, y_4)$, $\Psi_{4,5}(y_4, y_5)$, $\Psi_{5,6}(y_5, y_6)$, $\Psi_{6,1}(y_6, y_1)$, $\Psi_{1,2}(y_1, y_2)$, and $\Psi_{1,4}(y_1, y_4)$. By taking into account its junction tree, the following is proposed:

$$Z \times \mathrm{P}(\boldsymbol{Y}|\boldsymbol{X}) = \Psi'_A(y_1, y_5, y_6)\Psi'_B(y_1, y_4, y_5)\Psi'_C(y_3, y_4)\Psi'_D(y_1, y_2) \qquad [6.13]$$

with:

$$\Psi'_A = \Psi_{5,6}\Psi_{6,1} \qquad \Psi'_B = \Psi_{1,4}\Psi_{4,5}$$
$$\Psi'_C = \Psi_{3,4} \qquad \Psi'_D = \Psi_{1,2}$$

18 In other words, the cycles are longer than or equivalent to 4.

19 The factorization according to this order corresponds to a graph triangulation choice and the algorithm comes back to calculate the size of the graph tree.

In this manner, a factorization of P on the set of bands which correspond to the bands of its junction tree are evidenced. This tree factorization (the factors and junction tree decomposition) is the structure from which the evaluation of all marginal probability P can be efficiently done by dynamic programming.

What is essential to remember is that the tree obtained conserves the good properties of the probabilistic decomposition as well as that of a junction tree irrespective of the node at the root. In fact, due to this structure a certain vis-a-vis independence of variables elimination order. This note will make sense when the for message passing algorithm is exposed.

Finally, we note that when the original underlying P graph is a chain or a tree, the tree factorization is immediate.

6.5.4. *Inference in CRFs*

Once given this junction tree and the potential functions attached to these nodes, the message passing algorithm can be applied. Coming back to [6.13] definition of the probability is associated with the graph of Figure 6.4 to the right. To calculate the marginal probability in the first node$A$, for a given value of (y_1, y_5, y_6), the following sum is obtained:

$$\Psi'_A(y_1, y_5, y_6) \sum_{y_2, y_3, y_4} \Psi'_B(y_1, y_4, y_5)\Psi'_C(y_3, y_4)\Psi'_D(y_1, y_2)$$

that is always factorized according to the tree:

$$\Psi'_A(y_1, y_5, y_6) \sum_{y_4} \Psi'_B(y_1, y_4, y_5) \sum_{y_3} \Psi'_C(y_3, y_4) \sum_{y_2} \Psi'_D(y_1, y_2)$$

The same process as that described on page 199 is found here, which enables the efficient calculation of the marginal probability by dynamic programming. To be more explicit, the $m_{D\to B}(y_1) = \sum_{y_2} \Psi'_D(y_1, y_2)$ and $m_{C\to B}(y_4) = \sum_{y_3} \Psi'_C(y_3, y_4)$ factors which will be memorized can be evidenced in such a manner that:

$$\Psi'_A(y_1, y_5, y_6) \sum_{y_4} \Psi'_B(y_1, y_4, y_5) m_{C\to B}(y_4) m_{D\to B}(y_1)$$

The $m_{C\to B}(y_4)$ factor (even though $m_{D\to B}(y_1)$) can be interpreted as a *message* passed from the node$C$ to node$B$ (and from node$D$ to node$B$, respectively). The message, parametered by a value of y_4, is characterized in the form of a function dependent on y_4 and the calculation is realized at node$B$. Also, the calculation can be realized at node$B$ as a message sent to node$A$ with parameters (y_1, y_5):

$$m_{B\to A}(y_1, y_5) = \sum_{y_4} \Psi'_B(y_1, y_4, y_5) m_{C\to B}(y_4) m_{D\to B}(y_1) \qquad [6.14]$$

As a result, $P(Y_1 = y_1, Y_5 = y_5, Y_6 = y_6|\boldsymbol{X})$ is proportional to $\Psi'_A(y_1, y_5, y_6)$ $m_{B\to A}(y_1, y_5)$. What has been gained from this new formulation? Nothing for the moment, if no other marginal probability calculations are performed. By considering exactly the same approach for the other marginal probabilities, it can be concluded that the exchanged messages are identical, or that the emitters and receivers are simply reversed. Figure 6.5 shows the exchanged messages in different cases, and the node which calculates the marginal probability is placed at the root. To obtain an efficient calculation, the messages in the two senses must be calculated once for all. For this, a simple "there and back" is sufficient.

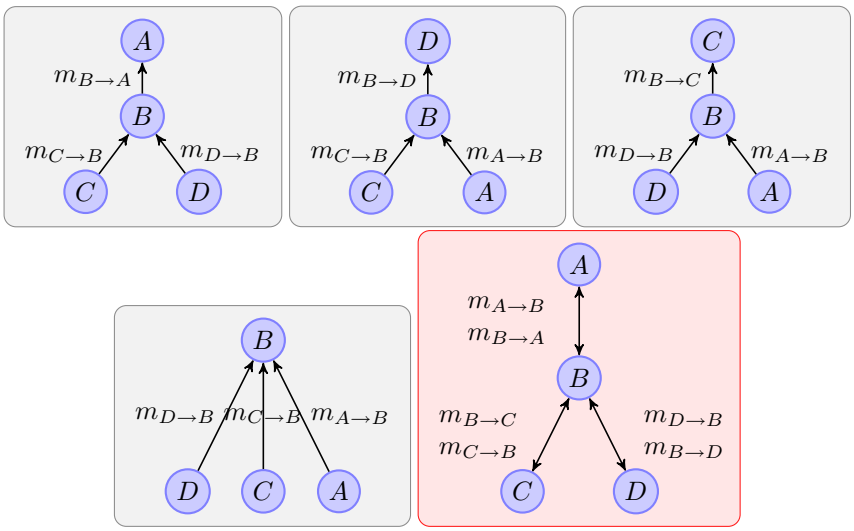

Figure 6.5. *Messages exchanged for the calculation of marginal probabilities (the last sketch show that a simple there and back enables the calculation of all the messages and therefore all the marginal probabilities)*

The similarity between this algorithm and that applied to the hidden Markov models can be noted. To perform this calculation, the significant property used is the distributivity of the multiplication in relation to the addition. As the multiplication is equally distributive in relation to the "maximum" operation (when only manipulating positive numbers), the "sum-product" version of the calculation showed here also extends to the "max-product" version which allows for the most probable labeling to be obtained.

6.5.5. *Inference algorithms*

To summarize, a general inference algorithm can now be given. Whether a junction tree $\mathcal{T}$ is associated with the graph on the output random variables $\boldsymbol{Y}$ of a Markov random field. For each node$v$ of $\mathcal{T}$, we note that $N(v)$ is the set of the neighboring v nodes. The variable in the nodes$v$ is noted along with $\boldsymbol{Y}_v$ and their realization is $\boldsymbol{y}_v$. The functions of the potential associated with each node$v$ of $\mathcal{T}$ are written as $\Psi'(\boldsymbol{y}_v)$. Finally, for the two v and v' nodes, $\boldsymbol{Y}_{v'} \setminus \boldsymbol{Y}_v$ represents the set of v' variables which are not in v and their realizations are written as $\boldsymbol{y}_{v'} \setminus \boldsymbol{y}_v$. What's more? $\boldsymbol{Y}_{v'} \cap \boldsymbol{Y}_v$ is the set of variables common to $\boldsymbol{Y}_{v'}$ and $\boldsymbol{Y}_v$.

The algorithm calculates for the first time all the messages between all the $\mathcal{T}$ nodesin two goes: one "there" and one "return". These consist of collecting the messages coming from the leaves and in turn resending the messages toward the leaves. The beginning of this recursion can be operated in any $\mathcal{T}$ node, therefore the choice is implemented by the *RootChoice* function. The values of the messages are memorized in the structures noted $m_{v \to v'}$ and indexed by the variables $\boldsymbol{y}_{v'} \cap \boldsymbol{y}_v$. If, for example, $\boldsymbol{y}_{v'} \cap \boldsymbol{y}_v$ contains two variables Y_j and Y_k then $m_{v \to v'}$ is a function with two arguments: it is therefore stocked in a matrix with the values of Y_j in the row and the values of Y_k in the column, where each cell contains the values of the $m_{v \to v'}$ message for the values of the corresponding variables. By commodity, we note that $m_{v \to v'}[\boldsymbol{y}_{v'} \cap \boldsymbol{y}_v]$ this value.

Function *ThereBack*($\mathcal{T}$):
  $r \leftarrow$ *RootChoice*($\mathcal{T}$)
  For all $v \in N(r)$
    There(r, v)
  For all $v \in N(r)$
    Back(r, v)

Function *There*(v, v'):
  For all $v'' \in N(v') \setminus \{v\}$
    There(v', v'')
  SendMessage(v', v)

Function *There*(v, v'):
  SendMessage(v, v')
  For all $v'' \in N(v') \setminus \{v\}$
    Back(v', v'')

Function *SendMessage*(v', v):
  For all $\boldsymbol{y}_{v'} \cap \boldsymbol{y}_v$
    $m_{v' \to v}[\boldsymbol{y}_{v'} \cap \boldsymbol{y}_v] = \sum_{\boldsymbol{y}_{v'} \setminus \boldsymbol{y}_v} \Psi'_{v'}(\boldsymbol{y}_{v'}) \prod_{v'' \in N(v') \setminus \{v\}} m_{v'' \to v'}[\boldsymbol{y}_{v''} \cap \boldsymbol{y}_{v'}]$

For all marginal probability of node$v$, for a given realization $\boldsymbol{y}_v$, is proportional to an expression which makes intervention in only one potential function in $\boldsymbol{y}_v$ and messages, which can be written as:

$$P(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{Y}_v = \boldsymbol{y}_v) \propto \Psi'_v(\boldsymbol{y}_v) \prod_{v' \in N(v)} m_{v' \to v}[\boldsymbol{y}_{v'} \cap \boldsymbol{y}_v]$$

6.5.6. *Training CRFs*

The problem with the training of Markov random fields consists of determining the parameters Θ of the model from labeled examples (see the **Pr2** on page 194). As for the entropy maximum models (see section 6.4.2), the classical manner of resolving it relies on the principle of the maximum likelihood , applied here to conditional distributions.

Let us consider a finite set of labeled data comprised of couples giving a realization of $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$S = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$$

The examples of S are *i.i.d.* [20] and the $\boldsymbol{y}^i$ that are linked to $\boldsymbol{x}^i$ according to a fixed probability but an unknown one which must be represented with the aid of a Markov random field of which the structure is known. Also we recall that the assumption of linking the parameters (see page 6.4.2) of the Markov random field allows for the consideration of the fields with the varied size and structure in S (of which all the elements are not of the same size), even if the properties of the underlying graph apply identically to each Y_j variable.

In a Markov random field with parameters Θ, it is most often chosen to optimize a loss function which corresponds to the log-likelihood, written as:

$$\begin{aligned} \ell(S; \Theta) &= \sum_{i=1}^{m} \log \mathrm{P}(\boldsymbol{y}^i | \boldsymbol{x}^i; \Theta) \\ &= \sum_{i=1}^{m} \sum_{c \in \mathcal{C}} \sum_{k} \theta_k f_k(\boldsymbol{y}^i_c, \boldsymbol{x}^i, c) - \sum_{i=1}^{m} \log Z(\boldsymbol{x}^i) \end{aligned}$$

The number of characteristic functions is often consequential, the number of parameters θ weighting these functions is too, and there is a significant risk of *over-learning*. To avoid this problem, a standardization term is very often added to the log-likelihood. For this, several possible solutions have been tested [PEN 04]. Often,

20 It must be noted that the variables in the $(\boldsymbol{x}^i, \boldsymbol{y}^i)$ field are not *i.i.d.*

$\sum_k \frac{\theta_k^2}{2\sigma^2}$ is searched for. Other approaches based on the ℓ_1 standardization are also possible and have the advantages of adding parsimony [SUT 06, SOK 09, LAV 10].

The choice of the parameterization in the form of an exponential model guarantees the optimum searched for the existing log-likelihood and is unique. However, searching for this optimum cannot be resolved analytically, and the solution must therefore be approached through optimization techniques. The development of the log-likelihood gives a difference between the two terms, one coming from the normalization coefficient Z. The derivations in relation to each θ_k of this expression evidences a familiar calculation of the maximum likelihood models for the families of exponential models [KLI 07, BER 96]. In fact, an interpretation in the following format appears:

$$\frac{\partial \ell(S;\Theta)}{\partial \theta_k} = \tilde{E}(f_k) - E(f_k)$$

where $\tilde{E}(f_k)$ represents the expectation of the characteristic function f_k according to the empirical distribution observed in the sample S and $E(f_k)$ is the expectation in the Θ parameters model. The expression is canceled out to the maximum likelihood and the calculation therefore comes back to evaluate the two functions $\tilde{E}(f_k)$ and $E(f_k)$. The first is simple to evaluate as it only counts how many times each characteristic function is realized in the sample. For the second, the calculation is generally impractical as all the possible values of variable $\boldsymbol{y}$ must be considered. The same factorization problem as for the inference problem occurs and the same solution applies.

In their founding article on the Markov random fields, Lafferty *et al*. proposed a training basis on the work of [PIE 97]. It is an adaptation to the Markov random fields of the *Improved Iterative Scaling* (IIS) algorithm, which uses a Newton optimization technique. Unfortunately, the convergence is slow and a large number of iterations is required for the convergence to occur. Using the results discussed in [MAL 02], [WAL 02] showed that the conjugated gradient methods, BFGS and L-BFGS converge much more rapidly than IIS. These strategies are therefore more often chosen in the majority of Markov random fields implementations.

However, other lanes have also been envisaged for training. In [COL 02, ROA 04], an approximation of the log-likelihood is realized by a technique close to perceptrons. At each stage, an example is presented and the gap between the weight of each characteristic observed is that predicted by the calculated model. An average of these gaps then corrects θ_k. The functional approximation of the gradient is also addressed in [DIE 04]. The authors proposed a representation of the potential functions by the sums weighted on the regression trees. These trees are constructed at the CART with the help of characteristic functions. The algorithms can not only be seen as a generation form (like combinations) of such functions, but also as a training getting further away from

the likelihood maximization. In fact, it is more a minimization criteria of a quadratic error that are applied. In the same spirit, other works proposed other criteria for the optimization problem. In this framework of the predicted structure, an adapted loss function can be considered. This is the case in [KAK 02], where the criteria used are based on the likelihood maximization of marginal probabilities (pseudo-likelihood). The nodesare not too penalized using this method. In numerous other works [TAS 04, COL 08], the maximization criteria of the margins are applied. A loss function is chosen, generally a Hamming type, and an optimization formulation according to this loss is defined. In a reformulation in a dual space, the marginal probabilities are found. As in many works on structure prediction [TSO 05], the passage to the ladder remains difficult but the same separability properties can be used and a difficulty in the number of reasonable variables can be obtained (for example, $n * |\mathcal{Y}|^2$ for the linear Markov random fields). Finally, the methods that are lent to support the vector machines (SMO, exponential gradient) have also been proposed to resolve the optimization problem in this dual space (see Chapter 4).

The fact is that there remains a learning problem in the classical sense, in which numerous variants can be envisaged while accounting for the environment. For example, the knowledge of the domain is sometimes diffuse and requires that several thousand or sometimes millions of characteristic functions are considered. The selection of probable characteristics is therefore primordial. Several authors have underlined the difficulties of realizing this *a priori* selection. However, a dynamic approach seems to be possible [MCC 03b]. The same objective is also obtained using a ℓ_1 standardization method: in [SOK 09], the authors show that at equal performances, the number of characteristic functions can be enormously reduced. The fact remains that an acceptable estimation of these models with numerous parameters require a large set of labeled examples. The availability of such examples is very rarely assured when numerous non-labeled examples are often easily accessible. The development of semi-supervised methods therefore seems to be natural. Unfortunately, this approach is even more complex in the case of discriminative models such as Markov random fields, as in the case of generative models. However, several authors have addressed this question [ZHU 03, LI 05, ALT 05, JIA 06]. For now, [SOK 08] evidenced conditions in which the input of the unlabeled data can become probable.

6.6. Conditional random fields and their applications

In section 6.5.6, we discuss the practical applications of CRF to linguistic engineering. The most simple Markov random fields, are introduced, those giving rise to the greatest number of implementations and with the hidden Markov models and the domains in which they are discussed, are presented. The more complex instances are also addressed and this chapter is concluded by a panorama of libraries available on the Internet which implement Markov random fields.

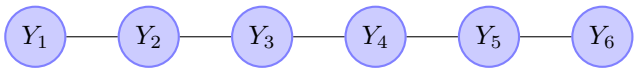

Figure 6.6. *Graph of a linear Markov random field*

6.6.1. *Linear conditional random fields*

Conditional random fields have been introduced by [LAF 01] for the segmentation and annotation of *sequences*, whether it may be a character or word chains. In this case, the structure of conditional independence graphs which is generally also chosen has a sequence format (or even chain format). These are known as *linear* Markov random fields, designated by a natural and total order between the random variables of each field and an independence of each annotation variable with those above its immediate successor and predecessor. For the example of page193 with six word, the graph between the annotation variables is shown in Figure 6.6. It is always important to understand well what introduces the structure of the graph associated with Markov random field. For example, a knowledge of the domain which says *the annotation of any word has more chance of being "EI" if the annotation of the first word is "EB"*, will be translated by a characteristic function and a dependence between each Y_i and Y_1 variable. In choosing a graph such as that in Figure 6.6, the usage of such characteristics becomes forbidden. Only the characteristic functions linking the value of the annotation of a word to the annotation of the following word (or previously since the graph is undirected) are enabled. We note, however, that knowledge of a long distance observation remains possible, as for example *the annotation of any word has more chance of being "EI" if the first word of the sentence is "London"*. We recall that in fact, the function arguments are designated by y_{i-1} and y_i, and the instantiation of the general CRF linear model with a size n gives the following format:

$$\mathrm{P}(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{i=2}^{n} \sum_{k=1}^{K} \theta_k f_k(y_{i-1}, y_i, \boldsymbol{x}, i) \right)$$

with:

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\left(\sum_{i=2}^{n} \sum_{k=1}^{K} \theta_k f_k(y_{i-1}, y_i, \boldsymbol{x}, i) \right)$$

In the case of linear Markov random fields, the maximal bands of the graph have a size of 2 (each couple of successive variables constitute such a band) and the optimal junction tree, here a chain, is evidently obtained without difficulty. For the inference problem, the message passing algorithm in this case is identical to the backward-forward algorithm (see Appendix A) used in the hidden Markov models.

The messages[21] going from the start of the chain toward variable i coincides with the habitual $\alpha_{y,i}$ and those going to the end of the chain toward variable i with $\beta_{y,i}$ habituals:

$$\alpha_{y,i} = \sum_{y'} \alpha_{y',i-1} \exp \sum_{k} \theta_k f_k(y', y, \boldsymbol{x}, i)$$

$$\beta_{y,i} = \sum_{y'} \beta_{y',i+1} \exp \sum_{k} \theta_k f_k(y, y', \boldsymbol{x}, i)$$

These new variables correspond to the non-normalized marginal probability of a partial labeling of the chain with Y_i labeled by y. Their definition is totally analog, but adapted to a particular case of a chain, to those of messages defined in equation [6.14]. By adding the definitions to the extremities of the chain $\alpha_{y,1}$ and $\beta_{y,n}$ which pose the basic case of the recurrence of the calculation of $\alpha_{y,i}$ and $\beta_{y,i}$, the marginal probabilities of each variable are obtained:

$$\mathrm{P}(Y_i = y|\boldsymbol{X} = \boldsymbol{x}) = \alpha_{y,i}\beta_{y,i}/Z(\boldsymbol{x})$$

$$Z(\boldsymbol{x}) = \alpha_{y,n} = \beta_{y,1}$$

The difficulty of the inference problem in this case of the order of $n|\mathcal{Y}|^2$.

In the case of linear Markov random fields, it can be seen that the calculation algorithms come back to those used for hidden Markov models. Going even further, it can be very simply verified that all modeled probabilities in such a model can also be done by a linear Markov random field.

6.6.2. *Relations between linear CRFs and hidden Markov models*

To show that the probabilistic distributions modeled by linear CRF include those described by the hidden Markov models, the simplest thing to do is to go from the one presented in this book in Appendix A, Figure A.2. The most significant aspect of this work is to transform the structure of this hidden Markov model from characteristic functions to a Markov random field. For this, three types of characteristic function are required:

– characteristic functions which translate the *initial states*: for each state u of the hidden Markov model which is an initial state ($u \in \{1, 2, 3\}$ in this example), the following characteristic function is defined as:

$$f_{k_1,u}(y_i, \boldsymbol{x}, i) = \begin{cases} 1 \text{ if } i = 1 \text{ and } y_1 = u, \\ 0 \text{ if not} \end{cases}$$

21 Messages can be written as $m_{i-1\to i}(y)$ and $m_{i+1\to i}(y)$, respectively.

– characteristic functions which translate the *transitions between the states*: for each state couple (u, u') in the hidden Markov model (in the example, $(u, u') \in \{1, 2, 3\}^2$), the following characteristic function is defined as:

$$f_{k_2,u,u'}(y_i, \boldsymbol{x}, i) = \begin{cases} 1 \text{ if } i > 1 \text{ and } y_{i-1} = u \text{ and } y_i = u' \\ 0 \text{ if not} \end{cases}$$

– characteristic functions which translate the *emission of states*: for each state u of the hidden Markov model ($u \in \{1, 2, 3\}$ in this example) capable of emitting an observation v ($v \in \{1, 2, 3\}$ in this example), the following characteristic function is defined as follows:

$$f_{k_3,u,v}(y_i, \boldsymbol{x}, i) = \begin{cases} 1 \text{ if } y_i = u \text{ and } x_i = v \\ 0 \text{ if not} \end{cases}$$

While going back to equation [6.10]:

$$\mathrm{P}(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{c \in \mathcal{C}} \exp\left(\sum_k \theta_k f_k(\boldsymbol{y}_c, \boldsymbol{x}, c) \right)$$

Instantiating the proposed calculation case in an example in Appendix A for the hidden Markov model, where $\boldsymbol{y} = 2312$ and $\boldsymbol{x} = 1211$ must be done. The c bands to be considered on the annotation variables Y_i for this calculation are as follows:

– the "unary" bands composed of a unique $Y_i = u$ variable: on each of these bands, the only non-zero characteristic functions are as follows:

- for variable Y_1, only, which has as a value $u = 2$: the function $f_{k_1,2}$,
- for each Y_i variable, initial or not, which has for value u whereas $X_i = v$: the characteristic function $f_{k_3,u,v}$,

– the "binary" bands composed of two successive variables Y_{i-1} and Y_i, with the respective values of u and u': the only feature which does not canceled out on each of the bands is $f_{k_2,u,u'}$.

On each band c (with the exception of band Y_1), a single characteristic function is 1 whereas all the others are canceled out: the value of the parameter θ_k associated with this unique characteristic function is not zero. The following simplified formula is thus obtained:

$$\mathrm{P}(\boldsymbol{Y} = 2312 | \boldsymbol{X} = 1211) = \frac{R}{Z(1211)}$$

with:

$$R = \exp(\theta_{k_1,2} + \theta_{k_3,2,1}) \exp \theta_{k_3,3,2} \exp \theta_{k_3,1,1} \exp \theta_{k_3,2,1} \exp \theta_{k_2,2,3} \exp \theta_{k_2,3,1} \exp \theta_{k_2,1,2}$$

For each index value k, known as $\lambda_k = \exp\theta_k$. After reordering to make the parameters of the first index k_3 appear at the end, the following is obtained:

$$R = \lambda_{k_1,2} \times \lambda_{k_2,2,3} \times \lambda_{k_2,3,1} \times \lambda_{k_2,1,2} \times \lambda_{k_3,2,1} \times \lambda_{k_3,3,2} \times \lambda_{k_3,1,1} \times \lambda_{k_3,2,1}$$

Now, all that required is the comparison of the calculation with that of page 405 to understand the interpretation which can be given to each of these λ_k parameters:

– $\lambda_{k_1,2}$ coincides with the probability that 2 is in an initial state, in other words the probability that characteristic function $f_{k_1,2}$ is satisfied;

– each $\lambda_{k_2,u,u'}$ coincides with the probability of passing from state u to state u', in other words the probability of satisfying characteristic function $f_{k_3,u,v}$.

So the parallel is complete, and it must be noted that the calculation performed in the hidden Markov model was only that of a joint probability. To obtain the corresponding conditional probability, the following must in fact be calculated:

$$\mathrm{P}(\boldsymbol{Y} = 2312|\boldsymbol{X} = 1211) = \frac{P(\boldsymbol{X} = 1211, \boldsymbol{Y} = 2312)}{P(\boldsymbol{X} = 1211)}$$

For reasons similar to those previously discussed, the calculation of $P(\boldsymbol{X} = 1211)$ in the hidden Markov model can easily be established and is similar to $Z(1211)$ in the Markov random field. We also note that the sketch of the hidden Markov model was not a complete graph, to assure the parallelism of the calculations, defining the characteristic functions corresponding to the absent connection by associating them to a $\theta = \lim_{x\to 0^+} \ln x = -\infty$ must be done.

Through this calculation, given the hidden Markov model it can be seen that it is very easy to fix the set of characteristic functions as well as the set of associated θ_k parameters which define exactly the same probabilistic distribution for all $(\boldsymbol{x}, \boldsymbol{y})$ data. This calculation even sheds an interesting light on the nature of the objects which intervene in the definition of the Markov random fields:

– the notion of characteristic function translates the knowledge of diverse nature, which correspond just as well to the *label structure* (the connectivity between the states of the hidden Markov model) as *the relationship between the label and data* (the emission of the hidden Markov model). But the characteristic functions for the above simulation are restricted in relation to all the expressivity authorized in a Markov random field. The possibility, in the Markov random fields, of defining characteristic functions which test properties on input data *no matter there position in the sequence* brings a supplementary degree of freedom, not enabled in the hidden Markov models. It is for this reason, when the characteristic functions are well chosen, the Markov random fields generally give the best results;

– the parameters θ_k which weight the characteristic functions are not probabilities: these are real numbers. The fact that, in the particular case of this example, it can be

expressed as $\theta_k = \ln \lambda_k$ where the λ_k are, themselves, probabilities, therefore comprised between 0 and 1, does not mean that they are always negative. However, the particular case of the weight worth ∞ is not significant here. In practice, their value varies from -10 to $+10$ and they are larger in absolute value of importance than the characteristic functions to which they are associated. They measure the *discriminant power* of these characteristic functions. This power is not necessarily correlated with their satisfaction probability, it is for this reason that it is not recommended to proceed to an *a priori* selection of characteristic functions based on the number of times they are verified.

6.6.3. *Applications of CRFs*

The main advantage of linear Markov random fields in relation to hidden Markov models, as illustrated in the previous section, is that they enable integration using their very diverse nature of knowledge characteristic functions. The relative importance of these characteristic functions for overall labeling will be determined during the learning phase of the parameters θ_k which weight them. Several experimentations tend to show the model to be just as efficient when based on a large number of characteristic functions. It is, therefore, beneficial to examine the different sources possible for these functions.

The characteristic functions which the linguist can regard as a priority are those which translate the explicit knowledge. If it is estimated, for example, that starting with a capital letter is a significant index for identifying the words which are part of the named entity, a characteristic function which exactly expresses this property can be defined (see the first characteristic function defined in section 6.4.2). Furthermore, if there is a list of words or group of words likely to fill certain fields of an extraction formula, it is also simple to transform them into characteristic functions. If L_1 is the first list of words of locations and L_2 is the list of words which follow these words of location (e.g. "Rio de Janeiro" "Rio" is part of L_1 whereas "de" and "Janeiro" are part of L_2). As these words are likely to fill "location" fields (label P for "place") of a database, the following characteristic functions can be defined as:

$$f_k(y_i, \boldsymbol{x}, i) = \begin{cases} 1 \text{ if } x_i \in L_1 \text{ (respectively } L_2) \text{ and } y_i = \text{“PB” (respectively “PI”)} \\ 0 \text{ if not} \end{cases}$$

The integration of regular expressions written by hand to characterize word configurations before or after an extraction zone inside the definition of characteristic functions can be envisaged, like in the following generic example where R is any regular expression:

$$f_{k'}(y_i, \boldsymbol{x}, i) = \begin{cases} 1 \text{ if } x_1 \ldots x_i \text{ satisfies } R \text{ and } y_i = \text{“PB”} \\ 0 \text{ if not} \end{cases}$$

It can, therefore, be seen that the knowledge present in an information extraction application conceived independently of all machine learning (see section 6.2.4) can be reinvested in the characteristic functions of a Markov random field. However, this knowledge is limited in number, and it is recommended to add others. In general, the majority of the characteristic functions are produced from available labeled examples in the learning set, using parametered patrons. A patron, in this context, is a structure which can be instantiated by browsing the examples. Imagine, for example, a patron with the following format:

$$f_{k''}(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} 1 \text{ if } x_{i-1} =? \text{ and } x_i =? \text{ and } x_{i+1} =? \text{ and } y_{i-1} =? \text{ and } y_i =? \\ 0 \text{ if not} \end{cases}$$

For each data couple $(\boldsymbol{x}, \boldsymbol{y})$ in the learning set and for each i position such that $1 < i < n$ in the sequence (assumed to be of size n), the interrogation point of this patron can be instantiated by the values present in example $(\boldsymbol{x}, \boldsymbol{y})$. A sentence with n words therefore gives rise to $n - 2$ distinct characteristic functions. A patron must of course respect the limitations which satisfy the characteristic functions: it can impose conditions on any $\boldsymbol{x}$ sequence value, but only on the values of sequence $\boldsymbol{y}$ inside the band identified by i. All the libraries which implement the Markov random fields (see section 6.6.5) offer the possibility of writing such patrons. McCallum and Li have been the first to use the linear Markov random fields to identify the diverse types of named entities, by mixing the characteristic functions issued from the learning data with others translating the knowledge collected on the Internet, in the form of proper lists [MCC 03a]. Other authors have adopted the same approach, mostly for biological named entities (genes, proteins, illness, etc.) and their relationships in the medical texts, which constitutes one of the most explored applications of information extraction [MCD 05, BUN 08].

The Markov random fields also offer the possibility of easily enchaining the learning of several successive models, by integrating the results obtained in the first models inside the characteristic functions of the following model. In [JOU 06, GIL 08], several combination propositions following this method are envisaged. When a first annotation, even a simplified and imperfect one, was obtained, this one can become the data of the following annotation, and therefore enter into the definition of its characteristic functions, even if they concern words far away from one another in the sentence. This strategy presents several advantages. It enables the simulation of taking into account the long distance dependencies between the annotation variables. It also enables the progressive introduction of several knowledge levels, translated in different characteristic function sets. Such an approach naturally envisages when the labels themselves can be decomposed into sub-labels hierarchically organized or into independent components that are able to be recombined after they go. Generally, the realization of several learnings (sequential or parallel) rather than a single one can be achieved, since each intermediate learning is more simple than that which consists of learning directly the more detailed labels. These approaches have been

developed in different applicative domains [ZID 10, TEL 10, BUN 08]. In [COH 05b], the coding of a set of labels in a binary language and independently learning each of the corresponding binary labelers has been proposed, by implementing classical strategies of error corrector codes. The algorithmic difficulty of the inference directly depends on the number of possible labels: reducing this number by coming back to simpler sub-labels therefore brings a gain in calculation time which can be significant.

Finally, we note that the use of linear Markov random fields on texts is not only limited to information extraction. All problems, which can be formulated as a sequence labeling problem, can be processed with a linear Markov random field. Sha and Pereira have, for example, reconsidered the idea of labeling the words of a text with "B", "I", or "O" but for spotting the boundaries of the non-recursive constituents, therefore cheaply producing what the Anglo-Saxons call a *shallow parser* [SHA 03]. *Chunks*, *clauses*, or even sentence boundaries [LIU 05] can therefore be spotted with Markov random fields. The labeling in *parts of speech* (POS) or in richer morpho-syntactic categories is the other large task with which the Markov random fields have been confronted from the beginning [LAF 01] and where they are immediately excelled. More recently, the benefit of performing word alignment tasks has been evidenced, with an indispensable preliminary stage of all the machine translation system [BLU 06, ALL 09] (see Chapter 7 of this book). Finally, the annotation models can also enable the labeling of portions of texts following semantic criteria (e.g. to identify the advertising messages on the Web, [SPE 10]). In this case, the units which are the object of labeling are the text sentences, namely larger portions of texts.

6.6.4. *Beyond linear CRFs*

In the theory of Markov random fields, there is no obligation to stay confined to a chain graph. Numerous works have tried to pass this limitation by considering more complex graphs, which enables us to take into account the knowledge of richer domains. We realize that this passage essentially as a difficulty problem, at the same time for the inference and training algorithm. We recollect the inference difficulty of the $n|\mathcal{Y}|^k$ order, where n is the number of variables (in other words the size of the data), and k is the size of the larger band in the junction tree.

For recognition tasks of named entities, and more generally for the domains where the annotation code of an overall processing of the sentence, the Markovian model of the linear Markov random fields is not always sufficient. For example, in this model it is not possible to avoid incoherent labeling, as the label indicates the middle of an entity (I for *Inside*) without its beginning (B for *Begin*) being signaled. This type of control can only be realized under a Markovian assumption if it is assumed that the dependencies between the labels are spaced at a certain *a priori* fixed limit. However, the larger the limit, the larger the graph bands will be, and therefore the higher the

risk of the calculation being unacceptable. A solution to this problem is proposed in [SAR 04] across the semi-Markovian random fields. In this work, these are text segments, in other words consecutive labeled word sequences. The words in the same sequence receive the same label and the dependence limitation is posed between the segments. In the *skip-chain* CRF [SUT 04] comprising another possible extension of the structure of the base model, by making the assumption of a hidden layer of labels, simultaneously learned to the final label. For other labeling tasks concerning more complex data such as images or semi-structured data, the knowledge of the domain also translates itself through large distance dependencies. In the case of images, the underlying graphs often have 2D or 3D grids enabling the labeling of a pixel (or a set of pixels) to its direct neighbors [KUM 03, HE 04]. To extract information from HTML pages or semi-structured documents, it is even possible to consider this data as linear text [PIN 03]. However, the most natural structure in this case is that of a tree whose nodescoincide with the tags not with the words. If the only dependencies taken into account between the annotation variables corresponding to the tree branches, whereas the size of the maximal band in the graph stays fixed at two and each path from the root to the leaf can be considered as a linear graph. The labeling of a tree is therefore not fundamentally more complex than a chain [COH 05a]. However, if we consider that the labeling of a nodealso depends on the labeling of its previous and subsequent brothers in the tree, then the graph structure becomes more complex: maximal bands with a size of three appears. Figure 6.7 shows two possible graphs associated with the same tree. Diverse experiments have shown that inclusion of a richer structure brings a gain for the labeling of HTML pages or tree banks [JOU 06, GIL 08, MOR 09]. In [GIL 08], it is also proved that all the probabilistic distributions expressible by the *runs* of a probabilistic tree PLC operating on binary trees can be simulated by a Markov random field of which the conditional independence graph has the structure of Figure 6.7 to the right. This result generalizes that discussed in section 6.6.2.

When the structure of the graph is too complex to enable an exact inference, approximative methods can be employed. In fact approximative inference algorithms on graphs with existing cycles. In this case, the initial graph is not triangulated, and variational methods [WAI 08] are applied. For example, in the *loopy belief propagation* algorithm, the message passing algorithm is used but with a slight modification: the messages are updated to each iteration by ignoring the cycles. Although it does not generally benefit from any convergence guarantee, the approach works very well in practice.

6.6.5. *Existing software*

To finish this location state, the implementations of the available Markov random fields on the Internet for all those that wish to test its efficiency must be discussed. The majority of these libraries limit themselves to linear Markov random fields, with the exception of XCRF, conceived for the annotation of trees in an XML format. The

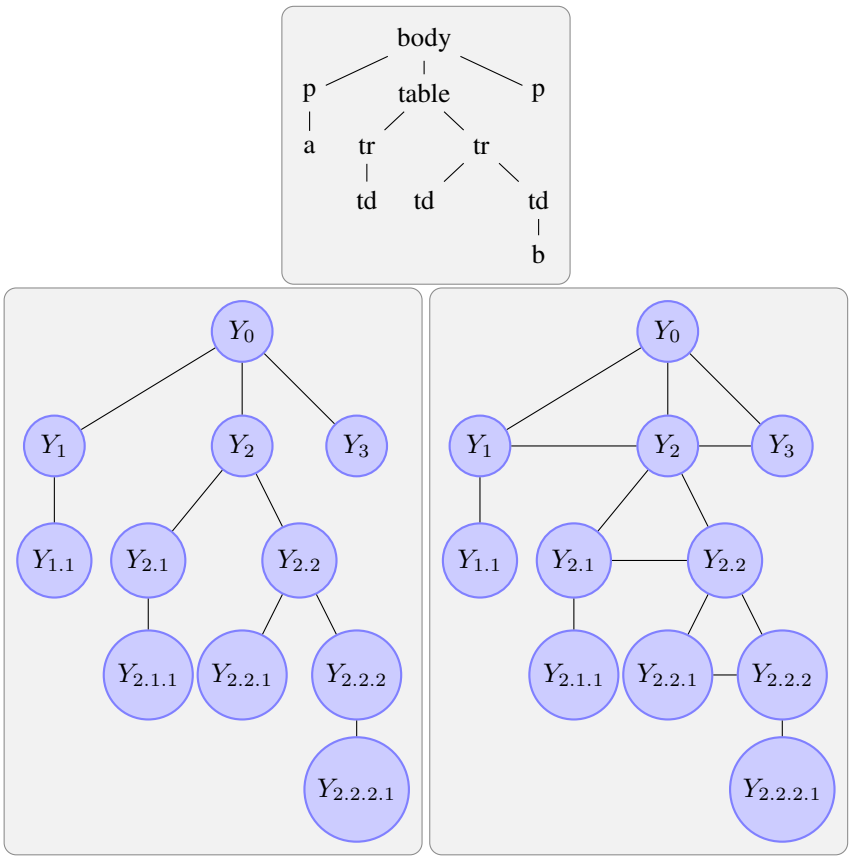

Figure 6.7. *Two possible graphs associate to the same XML/HTML tree*

format of expected data as well as the manner of specifying the characteristic functions strongly depend on the locations.

Libraries which implement Markov random fields. The list of freely available and most often used implementations, by signaling their main properties are listed here:

– *general libraries which include the Markov random fields among other tools*:
- MALLET[22], Java library that specialize in the statistical processing of texts,
- MINOR THIRD[23], set of Java classes for manipulating texts;

– *libraries that specialize in* CRF:

22 http://mallet.cs.umass.edu/

23 http://sourceforge.net/apps/trac/minorthird/wiki

- CRF library in Sarawagi Java[24] which also includes a location of the semi-Markovian CRFS,

- CRF++[25], an efficient implementation of C++ CRFS and POCKET CRF[26], a modified version of the same software,

- an optimized version which also realized the selection of characteristic functions for the "large CRFS", from where its name WAPITI[27] comes from: it is without doubt the most efficient implementation currently available[LAV 10],

- CARAFE[28], an OCaml implementation,

- FLEXCRFS[29], another C++ version which also includes a paralleled version known as PCRFS,

- CRFSUITE[30], a C++ implementation,

- CRF tool boxes for Matlab[31];

– *beyond linear CRFS*:

- XCRF[32], an implementation of Markov random fields for the trees which manages rich dependencies (right of Figure 6.7).

Regarding these implementations which go beyond the boundaries of CRFS, a FACTORIE [33] can be added, the project on which McCallum works, which aims to realize more general statistical inferences on graphs with more complex structures.

Citing to conclude software issued from the training of a CRF on the labeled data, therefore destined for a precise and specialized task for a particular language: by default this language is English but there are two pieces of software learned for a French corpus.

Specialized software for an applicative task:

– (chunker)[34] and a part of speech labeler (POS)[35] both learned according to PennTreebank with FLEXCRFS, at the University of Tohoku (Japan);

---

24 http://crf.sourceforge.net/

25 http://crfpp.sourceforge.net/

26 http://sourceforge.net/projects/pocket-crf-1/

27 http://wapiti.limsi.fr/

28 http://sourceforge.net/projects/carafe/

29 http://flexcrfs.sourceforge.net/

30 http://www.chokkan.org/software/crfsuite/

31 http://www.cs.ubc.ca/ murphyk/Software/CRF/crf.html or http://www.computervisiononline.com/software/conditional-random-field-crf-toolbox-matlab

32 http://treecrf.gforge.inria.fr/

33 http://code.google.com/p/factorie/

34 http://crfchunker.sourceforge.net/

35 http://crftagger.sourceforge.net/

– the labeler of named entities from Stanford[36], trained due to a corpus associating the data of diverse challenges (CoNLL, MUC-6, MUC-7 and ACE), under the GPL license;

– a tool for analyzing the structure of a document and identify the references and the citations it has[37], under the LGPL license and including and demonstrating a line interface;

– in the domain of bio-informatics, ABNER can be cited[38], a recognizer of biomedical named entities (gene names, proteins, cell types, etc.) and CONRAD[39], a gene predictor in the ADN structure;

– finally, for French, LIA_NE is used[40] as a recognizer of named entities acquired data from the ESTER challenge (transcription of radio conversations), and SEM[41], a segmenter-labeler learned from the French Treebank.

6.7. Conclusion

In this chapter, both a generic task (information extraction) and a family of statistical model (Markov random fields) are also capable of processing as efficiently as possible. In a case like this, the path which leads to the formulation of the task has quite a long resolution time. It is necessary to have several stages with diverse nature: reformulation of the task like an annotation problem, random variables field definition which model the domain, choice of a graph structure on these variables and of a model for expressing the conditional probability that links them, and most significantly the choice of features which describe the knowledge of the domain. Once this work is done, the algorithms capable of solving inference and learning problems generally have no need (except for a graph with an exotic structure) to be reinserted, such that the existing libraries stay performant.

The attraction of Markov random fields comes without a doubt from their capacity to *isolate the knowledge on which they are based* and their capacity to integrate in an overall model. It has therefore been shown that the different types of resources manually produced for processing an information extraction task can be reinvested in the characteristic functions of a CRF. In this hidden Markov models, this separation is less clear since a part of this knowledge resides in the structure of the models itself, and it in fact aims to *generate* the data it process. It must also be added that the model

36 http://nlp.stanford.edu/software/CRF-NER.shtml

37 http://wing.comp.nus.edu.sg/parsCit/

38 http://pages.cs.wisc.edu/ bsettles/abner/

39 http://www.broadinstitute.org/annotation/conrad/

40 http://pageperso.lif.univ-mrs.fr/ frederic.bechet/download.html

41 http://www.univ-orleans.fr/lifo/Members/Isabelle.Tellier/SEM.html

learned by a Markov random field is, at least in part, intelligible. The weighting of θ_k associated with the characteristic functions in fact translates their relative importance: a high absolute weighting value for characteristic functions therefore corresponds to a knowledge (positive or negative) which is important to take into account for resolving the task. By associating the symbolic local knowledge and overall statistical weighting, the Markov random fields combine the advantages of the two model families, which have competed against one another for a long time. It also links readability and robustness.

Conditional random fields have mostly proved themselves on a large number of problems, however, they are part of the "toolbox" of all good linguistic engineering practitioners.

6.8. Bibliography

[ALL 09] ALLAUZEN A., WISNIEWSKI G., "Modèles discriminants pour l'alignement mot-à-mot", *Traitement Automatique des Langues*, vol. 50, no. 3, p. 173-203, 2009, Numéro spécial, Apprentissage automatique pour le TAL.

[ALT 05] ALTUN Y., MCALLESTER D.A., BELKIN M., "Margin semi-supervised learning for structured variables", in *Advances in Neural Information Processing Systems 18 (NIPS)*, 2005.

[BER 96] BERGER A.L., PIETRA S.D., PIETRA V.J.D., "A maximum entropy approach to natural language processing", *Computational Linguistics*, vol. 22, no. 1, p. 39-71, 1996.

[BLU 06] BLUNSOM P., COHN T., "Discriminative word alignment with conditional random fields", in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, p. 65-72, July 2006.

[BUN 08] BUNDSCHUS M., DEJORI M., STETTER M., TRESP V., KRIEGEL H.-P., "Identifying gene and proteins mentions in text using conditional random fields", *BMC Bioinformatics*, vol. 207, no. 9, 2008.

[CAL 03] CALIFF M.E., MOONEY R., "Bottom-up relational learning of pattern matching rules for information extraction", *Journal of Machine Learning Research*, vol. 4, p. 177-210, 2003.

[CAR 07] CARME J., GILLERON R., LEMAY A., NIEHREN J., "Interactive learning of node selecting tree transducers", *Machine Learning*, vol. 66, no. 1, p. 33-67, January 2007.

[CHA 09] CHARNOIS T., PLANTVIT M., RIGOTTI C. CRÉMILLEUX B., "Fouille de séquence pour le TAL", *Traitement Automatique des Langues*, vol. 50, no. 3, p. 59-87, 2009, Numéro spécial, Apprentissage automatique pour le TAL.

[COH 05a] COHN T., BLUNSOM P., "Semantic role labelling with tree conditional random fields", in *CoNLL '05: Proceedings of The Ninth Conference on Natural Language Learning*, 2005.

[COH 05b] COHN T., SMITH A., OSBORNE M., "Scaling conditional random fields using error-correcting codes", in *43rd Annual Meeting of the ACL*, 2005.

[COL 02] COLLINS M., "Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms", in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1-8, 2002.

[COL 08] COLLINS M., GLOBERSON A., KOO T., CARRERAS X., BARTLETT P., "Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks", *Journal of Machine Learning Research*, vol. 9, p. 1775-1822, August 2008.

[COW 99] COWELL R., DAWID A., LAURITZEN S., SPIEGELHALTER D., *Probabilistic Networks and Expert Systems*, Information Science and Statistics, Springer-Verlag, 1999.

[DIE 04] DIETTERICH T.G., ASHENFELTER A. BULATOV Y., "Training conditional random fields via gradient tree boosting", in *Proceedings of the Twenty-first International Conference (ICML 2004)*, vol. 69 of *ACM International Conference Proceeding Series*, ACM, 2004.

[FIN 04] FINN A., KUSHMERICK N., "Multi-level boundary classification for information extraction", in *Proceedings of the European Conference on Machine Learning, Pisa, 2004*, p. 111-122, 2004.

[FRE 97] FREITAG D., "Machine Learning for information extraction in informal domains using grammaical inference to improve precision in information extraction", in *ICML Workshop on Automata Induction, Grammatical Inference and Language Acquisition*, 1997.

[FRE 00a] FREITAG D., "Machine learning for information extraction in informal domains", *Machine Learning*, vol. 39, no. 2/3, p. 169-202, 2000.

[FRE 00b] FREITAG D., KUSHMERICK N., "Boosted wrapper induction", in *AAAI/IAAI*, p. 577-583, 2000.

[GIL 08] GILLERON R., JOUSSE F., TOMMASI M. TELLIER I., "Conditional random fields for XML applications", Research Report no. RR-6738, INRIA, 2008.

[HAM 71] HAMMERSLEY J.M., CLIFFORD P., "Markov fields on finite graphs and lattices", 1971.

[HE 04] HE X., ZEMEL R., CARREIRA-PERPIÑÁN M.Á., "Multiscale conditional random fields for image labelling", in *Proceedings of CVPR 2004*, 2004.

[HOB 97] HOBBS J., APPELT D., BEAR J., ISRAEL D., KAMEYAMA M., STICKEL M., TYSON M., FASTUS: a cascaded finite-state transducer for extracting information in natural-language text, ROCHE E., SCHABES E. (eds), "*Finite State Language processing*", FASTUS: a cascaded finite-state transducer for extracting information in natural-language text, MIT Press, p. 383-406, 1997.

[JIA 06] JIAO F., WANG S., LEE C.-H., GREINER R., SCHUURMANS D., "Semi-supervised conditional random fields for improved sequence segmentation and labeling", in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[JOU 06] JOUSSE F., GILLERON R., TELLIER I., TOMMASI M., "Conditional random fields for XML trees", in *ECML Workshop on Mining and Learning in Graphs*, 2006.

[KAK 02] KAKADE S., TEH Y.W., ROWEIS S.T., "An alternate objective function for Markovian fields", in *Proceedings of the Nineteenth International Conference (ICML 2002)*, Morgan Kaufmann, p. 275-282, 2002.

[KLI 07] KLINGER R., TOMANEK K., "Classical probabilistic models and conditional random fields", Report no. TR07-2-013, Department of Computer Science, Dortmund University of Technology, December 2007.

[KOS 06] KOSALA R., BLOCKEEL H., BRUYNOOGHE M., DEN BUSSCHE J.V., "Information extraction from structured documents using-testable tree automaton inference", *Data & Knowledge Engineering*, vol. 58, no. 2, p. 129-158, 2006.

[KUM 03] KUMAR S., HEBERT M., "Discriminative fields for modeling spatial dependencies in natural images", in *Proceedings of NIPS 03*, 2003.

[LAF 01] LAFFERTY J.D., MCCALLUM A., PEREIRA F.C.N., "Conditional random fields: probabilistic models for segmenting and labeling sequence data", in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, p. 282-289, 2001.

[LAV 04] LAVELLI A., CALIFF M.-E., CIRAVEGNA F., FREITAG D., GIULIANO C., KUSHMERICK N., ROMANO L., "IE evaluation: criticisms and recommendations", in *Proceedings of Workshop Adaptive Text Extraction and Mining*, *American National Conference on Artificial Intelligence*, 2004.

[LAV 10] LAVERGNE T., CAPPÉ O., YVON F., "Practical very large scale CRFs", in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, p. 504-513, July 2010.

[LI 05] LI W., MCCALLUM A., "Semi-supervised sequence modeling with syntactic topic models", in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, AAAI Press / The MIT Press, p. 813-818, 2005.

[LIU 05] LIU Y., STOLCKE A., SHRIBERG E., HARPER M., "Using conditional random fields for sentence boundary detection in speech", in *43rd Annual Meeting of the ACL*, p. 451-458, 2005.

[MAL 02] MALOUF R., "A comparison of algorithms for maximum entropy parameter estimation", in *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, p. 49-55, 2002.

[MAN 99] MANNING C., SCHÜTZE H., *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, 1999.

[MAR 07] MARTY P., Induction d'extraction n-aire pour les documents semi-structurés, PhD thesis, Université Charles de Gaulle, Lille 3, 2007.

[MCC 03a] MCCALLUM A., LI W., "Early results for named entity recognition with conditional random fields", in *Proceedings of CoNLL 2003*, 2003.

[MCC 03b] MCCALLUM A., "Efficiently inducing features of conditional random fields", in *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, Morgan Kaufmann, p. 403-410, 2003.

[MCC 05] MCCALLUM A., "Information extraction: distilling structured data from unstructured text", *ACM Queue*, vol. 3, no. 9, 2005.

[MCD 05] MCDONALD R., PEREIRA F., "Identifying gene and proteins mentions in text using conditional random fields", *BMC Bioinformatics*, vol. 6, suppl. 1, no. 6, 2005.

[MOR 09] MOREAU E., TELLIER I., BALVET A., LAURENCE G., ROZENKNOP A., POIBEAU T., "Annotation fonctionnelle de corpus arborés avec des champs aléatoires conditionnels", in *actes de TALN 09*, 2009.

[NIG 99] NIGAM K., LAFFERTY J. MCCALLUM A., "Using maximum entropy for text classification", in *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999.

[PEN 04] PENG F., MCCALLUM A., "Accurate information extraction from research papers using conditional random fields", in *HLT-NAACL*, 2004.

[PIE 97] PIETRA S.D., PIETRA V.D., LAFFERTY J., "Inducing features of random fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, p. 380-393, 1997.

[PIN 03] PINTO D., MCCALLUM A., LEE X., CROFT W., "Table extraction using conditional random fields", in *SIGIR '03: Proceedings of the 26th ACM SIGIR*, 2003.

[POI 03] POIBEAU T., *Extraction automatique d'information*, Hermès, Paris, 2003.

[RAV 02] RAVICHANDRAN D., HOVY E., "Learning surface text patterns for a question answering system", in *Proceedings of the ACL conference*, 2002.

[ROA 04] ROARK B., SARACLAR M., COLLINS M., JOHNSON M., "Discriminative language modeling with conditional random fields and the perceptron algorithm", in *ACL*, p. 47-54, 2004.

[SAR 04] SARAWAGI S., COHEN W.W., "Semi-Markov conditional random fields for information extraction", in *Proceedings of NIPS*, p. 1185-1192, 2004.

[SAR 08] SARAWAGI S., "Information extraction", *Foundations and Trends in Databases*, vol. 1, no. 3, 2008.

[SHA 90] SHACHTER R.D., D'AMBROSIO B., FAVERO B.D., "Symbolic probabilistic inference in belief networks", in *AAAI*, p. 126-131, 1990.

[SHA 03] SHA F., PEREIRA F., "Shallow parsing with conditional random fields", in *Proceedings of HLT-NAACL*, 2003.

[SIT 04] SITTER A.D., CALDERS T., DAELEMANS W., *A Formal Framework for Evaluation of Information Extraction*, 2004.

[SOK 08] SOKOLOVSKA N., CAPPÉ O., YVON F., "The asymptotics of semi-supervised learning in discriminative probabilistic models", in *Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, vol. 307, ACM, p. 984-991, 2008.

[SOK 09] SOKOLOVSKA N., CAPPÉ O., YVON F., "Sélection de caractéristiques pour les champs aléatoires conditionnels par pénalisation l1", *Traitement Automatique des langues*, vol. 50, no. 3, p. 139-171, 2009, Numéro spécial, Apprentissage automatique pour le TAL.

[SPE 10] SPENGLER A., GALLINARI P., "Document structure meets page layout: loopy random fields for web news content extraction", in *DocEng '10: Proceedings of the 10th ACM Symposium on Document engineering*, ACM, New York, USA, p. 151-160, 2010.

[SUT 04] SUTTON C., ROHANIMANESH K., MCCALLUM A., "Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data", in *Proceedings*

of the Twenty-First International Conference on Machine Learning (ICML), p. 783-790, 2004.

[SUT 06] SUTTON C., MCCALLUM A., "An introduction to conditional random fields for relational learning", GETOOR L., TASKAR B. (EDS), *Introduction to Statistical Relational Learning,* MIT Press, 2006.

[TAS 04] TASKAR B., Learning structured prediction models: a large margin Approach, PhD thesis, Stanford University, 2004.

[TEL 10] TELLIER I., ESHKOL I., TAALAB S., PROST P., "POS-tagging for oral texts with CRF and category decomposition", *Research in Computing Science*, vol. 46, p. 79-90, 2010.

[TSO 05] TSOCHANTARIDIS I., JOACHIMS T., HOFMANN T., ALTUN Y., "Large margin methods for structured and interdependent output variables", *Journal of Machine Learning Research*, vol. 6, p. 1453-1484, 2005.

[WAI 08] WAINWRIGHT M.J., JORDAN M.I., "Graphical models, exponential families, and variational inference", *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, p. 1-305, 2008.

[WAL 02] WALLACH H., "Efficient training of conditional random fields", Master's thesis, University of Edinburgh, 2002.

[ZHU 03] ZHU X., GHAHRAMANI Z., LAFFERTY J.D., "Semi-supervised learning using Gaussian fields and harmonic functions", in *Proceedings of the Twentieth International Conference (ICML 2003)*, AAAI Press, p. 912-919, 2003.

[ZID 10] ZIDOUNI A., GLOTIN H., QUAFAFOU M., "Semantic annotation of transcribed audio broadcast news using contextual features in graphical discriminative models", in *Proceedings of CICLing 2010*, vol. 6008 of *LNCS*, Springer, p. 279-290, 2010.