Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–10

# Markov Logic Networks for Spoken Language Interpretation

Marie-Jean Meurs, Frédéric Duvert, Fabrice Lefèvre, and Renato De Mori

Université d'Avignon et des Pays de Vaucluse
Laboratoire Informatique d'Avignon (EA 931), F-84911 Avignon, France.
{marie-jean.meurs,frederic.duvert,fabrice.lefevre,renato.demori}
@univ-avignon.fr

## Abstract

In this paper, the use of Markov Logic Networks (MLN) is considered for application in spoken dialogue systems.

In spoken dialogues information that can be represented in logical form is often not explicitly expressed, but has to be inferred from detected concepts. Often, it is necessary to perform inferences in presence of incomplete premises and to get results with an associated probability. MLNs can be used for this purpose even in cases in which other known techniques llike CRF or Bayesian Networks cannot be easily applied.

Results on the inference of user goals from partially available information are presented using the annotated French Media corpus.

**Keywords:** spoken dialogue system, spoken language understanding, markov logic network, probabilistic logic

## 1 Introduction

In early Spoken Language Understanding (SLU), interpretations were carried on by binding variables and instantiating objects based on Automatic Speech Recognition (ASR) results [Walker *et al.* (1977)]. More recently, techniques for automatic learning from corpora were introduced in which interpretation is seen as a classification or a translation process. Classifiers and translators are automatically trained. As a first step, a surface semantic representation is generated. The elements of such representations are often called semantic or concept tags. Concept tags may represent relation names, variable types and values, function names. In a second step, composition of tags is performed into structures. Composition has to satisfy structural descriptions that are commonly expressed in logic form (see [Brachman (1978)]). Thanks to this representation, assertions not present in a natural language message can be obtained by inference. Composition knowledge can be learned from examples but can also be compiled by experts because semantic structures have precise definitions in terms of their components.

In a recent paper [Damnati *et al.* (2007)], a dialogue system has been described. It includes semantic knowledge represented in logic form following the KLONE model [Brachman (1978)]. Logic knowledge has been integrated in a probabilistic finite-state model.

When SLU systems are used in a telephone service, most often their output is processed with the objective of executing an action to satisfy one or more user goals.

In applications such as information seeking dialogues, there are different types of actions that can be performed, based on concept tags which have been detected and possibly been associated with values. Examples of actions are inferences for asserting new facts not explicitly expressed in the spoken message. System actions, like the access to a database, are performed when enough information is available, e.g, for submitting a request. Requests to users are issued when the dialogue manager considers that it needs more information in order to satisfy a user goal.

While inferences can be performed in a logic framework, other types of actions are described by precondition, action, post-condition rules which are not necessarily described by logic formulae. These actions can be executed only if preconditions are asserted as true. Action results are represented by post-conditions. Asserting preconditions for performing actions can be considered as intermediate goals useful for reaching a final goal which satisfies what the system considers a well understood user request.

In this framework, the role of SLU is to hypothesize the truth of preconditions for possible actions. Generation of hypotheses is affected by degrees of imprecision because SLU knowledge is often imperfect and the transcription of user utterances in terms of words may contain errors. Furthermore, information may be provided by users in different dialogue turns. Thus, in a given dialogue turn, there may be several possible goal hypotheses which are all affected by a degree of uncertainty due to the fact that only some elements of the goal description (e.g. preconditions for an action) have been hypothesized and the hypotheses are uncertain.

In order to make decisions about future dialogue actions, it is useful to estimate a probability that an hypothesized goal or sub-goal is part of the user intentions. Interesting approaches have been proposed for combining the expressive power of first order logic and the possibility of assigning probabilities to imprecise or incomplete inferences using exponential methods.

In this paper, the possibility of using Markov Logic Networks (MLN) in SLU systems is discussed and a probabilistic model for predicting user goals is proposed. Section 2 introduces the use of probabilistic logic for composing concept tags into structures and predicting user goals. Section 3 provides a background of MLNs. Section 4 describes experimental results.

## 2 On the use of probabilistic logic for semantic composition

Essential definition of logic formulas used in the following can be found in books [Nilsson (1980)] [Nilsson (1986)].They are now briefly reminded. Logic formulae are used for structural description of semantic objects and potential goals. They are part of the domain independent and the domain dependent semantic knowledge of an application. Their computational repository is a *knowledge base (KB)*. Formulae contain variables which are bound to constants and may be typed. An object is built by binding all the variables of a formula or by composing existing objects.

Formulae return truth and are constructed using *constants* which represent objects and may be typed, *variables*, *functions* which are mappings from tuples of objects to objects and *predicates* which represent relations among objects. An *interpretation* specifies which objects, functions and relations in the domain are represented by which symbol. A *term* is an expression representing an object (it can be a constant, a variable or a function). An *atomic formula* (or *atom*) is a predicate applied to a tuple of terms.

A *formula* is recursively constructed from atomic formulae using logical connectives (conjunction, disjunction, negation, implication, equivalence) and quantifiers (universal, existential). Formulae of KB are implicitly conjoined, so KB is a large formula.

*A* ground term is a term containing no variables. A *ground atom* (or *ground predicate*) is an atomic formula all of which arguments are ground terms. Atoms are grounded when values for variables are found and associated with predicates. Based on ground atoms, inferences can be performed in the KB to instantiate objects. Hypotheses about functions and instantiated objects are written into a Short Term Memory (STM).

In SLU, interpretations are carried on by binding variables and instantiating objects based on ASR results and inferences performed in the KB. A *possible world* assigns a truth value to every possible ground atom. A formula is *satisfiable* if and only if there exists at least one world in which it is true.

The basic *inference problem* consists in determining whether KB $\models$ F which means that a formula F is true in all worlds in which KB is true. The semantic knowledge used in [Damnati *et al.* (2007)] is represented in logical form by defining entities with roles, specifying the type of values the roles can take and a structural description expressing relations among roles and their fillers [Brachman (1978)].

A practically useful class of user goals consists in requests which can be fulfilled by instances of structures and roles represented by ground atoms. As dialogue progresses, some predicates are grounded by the detection of predicate tags, property tags and values. Such detection is made by the interpretation component. Other predicates are grounded as an internal result of inference. A user goal is asserted when all its components are grounded and asserted true.

The proposed approach is illustrated by the example of a user goal consisting in a reservation of an hotel in a town for some dates. Such a request implies the grounding of the predicates represented by the premise of an inference rule that corresponds to the structural description of the request:

$\texttt{town(t)} \wedge \texttt{hotel(h)} \wedge \texttt{date(d)} \wedge \texttt{reservation(i)} \wedge \texttt{resHotRelated(i,h)} \wedge \texttt{resDateRelated(i,d)} \wedge \texttt{isLocatedIn(h,t)} \Rightarrow \texttt{requestReservation (h,t,d)}$.

The rule contains unary predicates expressing the roles of the request and binary predicates expressing the necessary relations among roles.
A simple example of implications is made by the following formula grounded by constants A,B and G of the following sentence: *I have to fly from A to B and I would like to know about ground transportation (G) at the destination.*

`move(A,B)` $\wedge$ `queryService(G,destination)` $\Rightarrow$ `queryService(G,B)`
where `queryService(z,y)` means query about `z` at location `y`.

Once the inference is completed, a request can be submitted to a database that will produce an answer. If the answer is negative, the system should find out which ground atoms prevented from performing the transaction in the database. If the answer is positive, then it can be used to instantiate a post-condition structure by assigning values to all its roles.

Grounding the predicates of a premise for asserting a goal is a process that goes through a sequence of states. Let $\Gamma_{i,k}$ be the content of the STM used for asserting the predicates grounded at the $k^{th}$ turn which are part of the premise for asserting the $i^{th}$ goal. Let $G_i$ be an instance of the $i^{th}$ goal asserted after grounding all the predicates in the premise.

$\Gamma_{i,k}$ can be represented by a composition from a partial hypothesis $\Gamma_{i,k-1}$ available at turn $k-1$, the machine action $a_{m,k-1}$ performed at turn $k-1$ and the hypothesis $x_{i,k}$ about a semantic component $\gamma_{i,k}$ generated at turn $k$, based on evidence provided by acoustic features $Y_k$, i.e.: $\Gamma_{i,k} = \chi(x_{i,k}, a_{m,k-1}, \Gamma_{i,k-1})$.

Let $S_k(G_i)$ be the information state representing the hypothesis that $\Gamma_{i,k}$ is a partial grounding of the premise for asserting $G_i$. State probability can be written as follows: $P(S_k(G_i)|Y) = P(G_i|\Gamma_{i,k})P(\Gamma_{i,k}|Y)$ where $P(G_i|\Gamma_{i,k})$ is the probability that $G_i$ is the type of goal that corresponds to the user intention given the ground predicates in $\Gamma_{i,k}$. Notice that $\Gamma_{i,k}$ not only contains the ground atoms which are premises for asserting $G_i$, but also the portion of the KB which has been used to assert these predicates.

Predicates in a structural description like `isLocatedIn(h,t)` express the fact that the user did not simply mention a town and a hotel, but indicated that the goal is about *a hotel h in a town t*. It is possible, in fact, to mention a hotel and a town in a discourse with a different relation, like in the sentence *I read about town A in my hotel room.* The fact that a user asks about *a hotel h in a town t* does not necessarily implies that the relation `isLocatedIn(h,t)` will be asserted after a database request.

It is useful, for deciding about dialogue actions, to define dialogue states and compute their probabilities. Dialog states may represent successive phases of an inference process which may use incomplete or imprecise implication rules. Probabilities of states can be used to define a belief of a dialogue system and justify the need for the computation of $P(S_k(G_i)|\Gamma_{i,k})$. To that extend, the MLN seems an appropriate framework to derive probabilities from logically formulated structures.

## 3 Markov Logic Network

A Markov logic network (MLN) is a probabilistic knowledge representation over a finite domain of objects. MLNs provide a simple although efficient way to specify very large Markov networks and are able to incorporate a wide range of domain knowledge. But, more than that, a major asset of MLNs is their ability to handle uncertainty, i.e. to tolerate imperfect and contradictory knowledge and thus reducing brittleness. More details can be found in [Richardson and Domingos (2006)]. Some definitions are given in the following.

### 3.1 Markov network

A Markov network is a model for the joint distribution of a set of random variables $\mathcal{X}$ [Pearl (1988)]. $X$ is a Markov network (or a Markov random field) iff the conditional probability of a random variable is only function of its neighbours in the k-cliques of dependencies. Formally, a Markov network consists of:

- an undirected graph $G = (V, E)$, where each vertex $v \in V$ represents a random variable in $\mathcal{X}$ and each edge $(u, v) \in E$ represents a dependency between the random variables $u$ and $v$.
- a set of potential functions $\phi_k$, one for each clique $k$ in $G$. Each $\phi_k$ is a mapping from possible joint assignments (to the elements of $k$) to non-negative real numbers.

From the Markov network, the joint distribution of $X$ is derived as:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$$

where $x_{\{k\}}$ is the state of the random variables in the $k$th clique and the partition function $Z$ (normalizing constant) is: $Z = \sum_{x \in \mathcal{X}} \prod_k \phi_k(x_{\{k\}})$.

### 3.2 Markov Logic Networks

A first-order KB defines a set of possible worlds with hard constraints. A world violating a constraint has a null probability. A distinctive feature of MLNs is their ability to smooth the constraints. Whenever a world is in contradiction with a formula, its probability is lowered, not zeroed. And globally the fewer formulae a world violates, the more probable it is. A weight is associated to each formula reflecting how strong the constraint it represents is: the higher the weight, the greater the difference in log probability between a world that satisfies the formula and one that does not. At the limit of infinite weights, the MLN is a logical model (the probability distribution tends to converge to a uniform distribution over the worlds satisfying the $KB$).

Formally, an MLN $L$ is a set $\{(F_i, w_i)\}$ with $F_i$ a formula in first-order logic and $w_i$ a real number. Together with a finite set of constants $C = \{c_1, c_2, ..., c_{|C|}\}$,

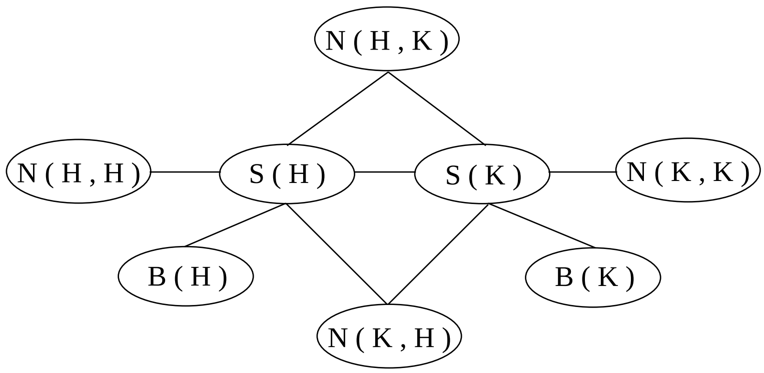
$\{(F_i, w_i)\}$ defines a Markov network $M_{L,C}$ (example: see Figure 1). This network has one node for each possible grounding of each predicate appearing in $L$. The node value is 1 if the ground atom is true, and 0 otherwise. The weight of the feature in the potential function is given by the $w_i$ associated with $F_i$ in $L$. The weight of a formula $F$ is the log likelihood difference between the worlds where $F$ is true and where it is false. Unfortunately, if $F$ shares variables with other formulae, it is hardly possible to have those formulas' trueness unchanged while reversing $F$'s. The one-to-one correspondence between formulae weights and probabilities is then lost. This is the reason why a learning strategy is used to derived the formulae weights. The algorithm provided in [Richardson and Domingos (2006)] uses a Monte-Carlo maximum likelihood estimator and the limited-memory BFGS algorithm.

To illustrate the use of MLN, an MLN providing a model to describe relations in the hotel reservation field can be built. Here, the predicate $N(h, k)$ means "hotel $h$ and hotel $k$ are neighbours", $S(h)$ means "$h$ is near the sea" and $B(h)$ means "$h$ offers beach leisure activities". Example of constraints is proposed in Table 1:

TABLE 1: *Proximity of the sea*

| English | First order logic | Clausal form | Weight |
|---|---|---|---|
| If two hotels are neighbours, either both are near the sea or neither does. | $\forall h \forall k\ N(h,k) \Rightarrow (S(h) \Leftrightarrow S(k))$ | $\neg N(h,k) \vee S(h) \vee \neg S(k)$<br>$\neg N(h,k) \vee \neg S(h) \vee S(k)$ | 1,5<br>1,5 |
| Proximity of the sea entails beach leisure activities | $\forall h\ S(h) \Rightarrow B(h)$ | $\neg S(h) \vee B(h)$ | 2 |

FIGURE 1: *Grounded Markov network obtained by applying the two formulae of Table 1 to the constants $H$ and $K$*




An MLN and different constant sets produce different Markov networks but all have regularities in their structure and parameters. For instance, all groundings of

the same formula have the same weight. Each state of $M_{L,C}$ stands for a possible world. The probability distribution over possible worlds of an $M_{L,C}$ is given by: $P(X = x) = \frac{1}{Z}exp\Big(\sum_i w_i n_i(x)\Big) = \frac{1}{Z}\prod_i \phi_i(x_{\{i\}})^{n_i(x)}$ with $n_i(x)$ the number of true groundings of $F_i$ in the world $x$, $x_{\{i\}}$ the truth values of the atoms appearing in $F_i$ and
$\phi_i(x_{\{i\}}) = e^{w_i}$.

As an example, let us consider the MLN containing the second formula of Table 1 and the set of constants $C = \{H\}$. Four worlds are possible: $\{S(H), B(H)\}$, $\{\neg S(H), B(H)\}, \{S(H), \neg B(H)\}, \{\neg S(H), \neg B(H)\}$ . From the last equation, we obtained $P(\{S(H),\ \neg B(H)\}) = 1/(3e^2 + 1)$ . The probability of each of the other three worlds is $e^2/(3e^2 + 1)$.

### 3.3 Inferences

The inference mechanism is generalized to any formula in the MLN. The probability of the formula $F_1$, given that the formula $F_2$, is provided by: $P(F_1|F_2, L, C) = \frac{\sum_{x\in\mathcal{X}_{F_1}\cap\mathcal{X}_{F_2}} P(X=x|M_{L,C})}{\sum_{x\in\mathcal{X}_{F_1}} P(X=x|M_{L,C})}$ such that $L$ is an MLN, $C$ is a set of constants appearing in $F_1$ and $F_2$ and $\mathcal{X}_{F_i}$ is the set of worlds where $F_i$ appears[1]. An approximation is obtained using a MCMC (Monte Carlo Markov Chain) algorithm: it considers only worlds where $F_2$ is satisfied and counts the number of samples in which $F_1$ holds. Richardson & Domingos [Richardson and Domingos (2006)] provides an efficient variant using Gibbs distributions and a local search algorithm (*MaxWalkSat*).

## 4 Experiments

In order to evaluate the possibility of using MLNs for computing $P(G_i|\Gamma_{i,k})$, some preliminary experiments were carried out on the annotated French Media dialogue corpus [Bonneau-Maynard *et al.* (2006)]. The Media corpus has been recorded using a wizard of Oz system simulating a vocal tourist information and hotel booking phone server. The corpus accounts 1257 dialogues from 250 speakers and is on the order of 70 hours of transcribed dialogues. The Media training corpus is conceptually rich (more than 80 basic concepts) and manually transcribed and annotated. In the latter experiments, the reference concept annotations will be used. To give an idea, a minimum loss of 20% in understanding error rate is observed with an automatic extraction of the concept sequences [Lefèvre (2007)].

In the reference annotations, only the user messages are annotated with basic concept tags, associated values and generic references. Relations and compositions are not available. The wizard system made inferences about user intentions and

[1]Computing $P(F_1|F_2, L, C)$ is tractable only for small domains. Probabilistic inference is #P-complete and logical inference is NP-complete [Richardson and Domingos (2006)].

proposes solutions based on what it was supposed to find in a database.

Our first set of experiments was limited to the computation of $P(G_i|\Gamma_{i,k})$ for hotel room reservation with constraints on dates, rooms and facilities. An example of such a goal is a request about a hotel in a city, offering some services such as tennis court. Its logical formulation is:
`requestReservation(hotel?, town_Paris, service_Tennis, days_Apr07_13-15).`
It may notice that some groundings have to be inferred. For instance, the user may ask for a double-bed room without mentioning that it is in a hotel even if this sounds implicit. Many relations of that kind, necessary for asserting a goal, are not annotated and have to be inferred automatically.

Not many dialogues are required for training the MLN used for computing $P(G_i|\Gamma_{i,k})$ because the frequencies of occurrences of values are already accounted for in $P(\Gamma_{i,k}|Y)$. Instead, it is crucial to elaborate a well calibrated KB by considering a set of dialogues encompassing all important aspects of the domain knowledge and by further generalizing this knowledge with human expertise.

User goals are not annotated in the MEDIA corpus. In telephone services, user goals can be inferred from the actions performed by human operators. By analyzing these actions, logical relations have been derived. With the analysis of these relations, representations of user goals have been derived. They are expressed by implications whose premises are conjunctions of predicates representing relations. The most complex relations are not annotated and have to be inferred for hypothesizing user goals. An important dialogue task is to infer user goals as the dialogue progresses. Often, premises of inferences do not have all the required predicates asserted. Nevertheless, using an MLN, it is possible to compute a probability of the inference of a goal based on partially available knowledge.

If goals cannot be inferred because the facts asserted in the STM are not sufficient, the verification of a necessary but not asserted relation can be required and performed. For example, the detection of an hotel and a town may trigger the verification of the relation (assertion of predicate) `isLocatedIn(h,t)`. Verification can be performed by asking the user a question or executing a classifier that outputs the truth of the predicate as a function of pertinent information already asserted and values for functions of words in the spoken message. MLN can also be used to compute the probability that verifying a relation will be allowing a pertinent inference in future processing. This possibility has not been evaluated yet.

Experiments have been performed on the computation of a user goal probability, given incomplete information asserted in the STM. This probability will be used with the acoustic evidence of the components to obtain the probability of an information state.

A test for the recognition of user goals was performed using 50 MEDIA dialogues having 888 user dialogue turns and 1108 user goals. Basic concepts related to predicates such as `hotel,town,service,roomtype,date` and `reservation`, were automatically selected from context-independent annotations. When a basic concept appears, the corresponding atoms and formulae are grounded. If possible, basic compositions with other concepts ( collected on the actual and previous turns) are grounded. All the ground atoms are collected in the STM. Learning and inference computations are performed using *Alchemy* [Kok *et al.* (2005)].

Test results are represented in Table 2. User goals in 15 dialogues were completely detected. For the others some goals were not detected. Reasons are also shown in Table 2. The main reason for not detecting goals was the absence of context-dependent relations. From the results, a strategy for scheduling verification of complex relations can be derived.

TABLE 2: Goal retrieval as a function of the number of user turns (total of 1108 goals).

| # Goals found | | # Goals not found (last turn) | |
|---|---|---|---|
| Turn 4 | 244 (22%) | Missing basic concepts | 197 |
| Turn 6 | 302 (27%) | Missing compositions | 53 |
| Last turn | 855 (77%) | Undetermined cases | 3 |

## 5 Conclusion

In this paper, a first step for using probabilistic logic in SLU is introduced. This step is necessary for retrieving unexpressed or undetected concepts and their relations and for dealing with coreference .

MLNs appear to be an efficient model, providing inferences even from incomplete or inconsistent data. Furthermore, they can deal with cases for which Conditional Random Fields and Bayesian Networks are not applicable [Richardson and Domingos (2006)].

Preliminary results on user goal inference have shown that the approach is viable and have suggested future research work, like detecting situations in which additional information has to be elicited in order to make possible the assertion of predicates necessary to infer conditions for performing system actions.

## References

H. Bonneau-Maynard, C. Ayache, F. Béchet, A. Denis, A. Kuhn, F. Lefèvre, D. Mostefa, M. Quignard, S. Rosset, C. Servan, and J. Villaneau (2006), Results of the French Evalda-Media evaluation campaign for literal understanding, in *LREC*, pp. 2054–2059, URL `ftp://tlp.limsi.fr/public/lrec06media.pdf`.

R. J. Brachman (1978), On the epistemological status of semantic networks, *NASA STI/Recon Technical Report N*, 78:31337–+.

G. Damnati, F. Béchet, and R. De Mori (2007), Spoken Language Understanding strategies on the France Telecom 3000 Voice Agency corpus, in *IEEE ICASSP*.

S. Kok, P. Singla, M. Richardson, and P. Domingos (2005), The Alchemy system for statistical relational AI, in *Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA*, URL `http://alchemy.cs.washington.edu`.

F. Lefèvre (2007), Dynamic Bayesian Networks and Discriminative Classifiers for Multi-Stage Semantic Interpretation, in *IEEE ICASSP*.

N. Nilsson (1980), *Principles of Artificial Intelligence*, Tioga Publishing Company, Palo Alto, CA.

N. Nilsson (1986), Probabilistic logic, *Artificial Intelligence*, 28:71–87.

J. Pearl (1988), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann publishers, San Mateo, California.

M. Richardson and P. Domingos (2006), Markov Logic Networks, *Machine Learning*, 62:107–136.

D. Walker, W. Paxton, B. Grosz, G. Hendrix, A. Robinson, J. Robinson, and J. Slocum (1977), Procedures for Integrating Knowledge in a Speech Understanding System, in *IJCAI*, pp. 36–42.