Short Note on Conditional Random Field model:

Conditional Random Fields [Edit]

12 min read

Introduction

Conditional Random Fields are **probabilistic graphical models for sequential or structured data**. They allow us to perform classification taking into account the context delivered by the sequence. We talk about a *structured prediction*, where segments are assumed to be related with each other.

By doing so, valuable contextual information, which would be lost in individual classifications, can be given to the model. For example, words in a sentence are grammatically connected: after an adjective it is more likely to find a noun than a verb. And this hint can be used to label the noun *books* in the sentence "The woman carefully carried the two red books".

How the algorithm works

Let us explain the construction of a CRF for a **part-of-speech (POS) tagging** task. Let x be the input sentence of length N: $x = (x_1, ..., x_N)$. Let x_i be the word at position i. Let y be the label-vector of the entire sentence and y_i the label for word x_i.

The first part of the problem deals with the data representation or **feature extraction**. For this matter, a set of m feature functions f_j is defined. Each function f_j in the set is applied to every word x_i in the sentence. We write $f_j(y,x,i)$ to denote the dependency of the function on vectors x and y, and the application at position i. Secondly, **the model is trained** in order to learn weights w_j on them. Next, weighted features are added up across all words and functions. Outputs are **scores** s for each possible vector y:

$$s(y|x) = \sum_{j=1}^{m} \sum_{i=1}^{N} w_j f_j(y, x, i)$$

Finally, scores are transformed into **probabilities**:

$$p(y|x) = \frac{e^{s(y|x)}}{\sum_{y'} e^{s(y'|x)}}$$

Since the feature values f can depend on not just the input x but also on output y, we can make coherent predictions. If each label was predicted independently, predictions might often not make sense as a whole. However, because of the same reason, this makes the training and prediction process much more computationally intensive. The value of f depends on y, and hence in the most general setting, every possible value of the sequence y needs to be tried for the score to be calculated. In-order to circumvent this computational issue, we often make f depend on small parts of y, so that we only need to try all possible values of that part of y. (We'll see a concrete example below).

Module 4:

1. **Applications of WSD:**
   Word sense disambiguation (WSD) is applied in almost every application of language technology.
   a. **Machine Translation**
      Machine translation or MT is the most obvious application of WSD. In MT, Lexical choice for the words that have distinct translations for different senses, is done by WSD. The senses in MT are represented as words in the target language. Most of the machine translation systems do not use explicit WSD module.

   b. **Information Retrieval (IR)**
      Information retrieval (IR) may be defined as a software program that deals with the organisation, storage, retrieval and evaluation of information from document repositories, particularly textual information. The system basically assists users in finding the information they require but it does not explicitly return the answers of the questions. WSD is used to resolve the ambiguities of the queries provided to the IR system. As like MT, current IR systems do not explicitly use WSD modules and they rely on the concept that the user would type enough context in the query to only retrieve relevant documents.

   c. **Text Mining and Information Extraction (IE)**
      In most of the applications, WSD is necessary to do accurate in of text. For example, WSD helps intelligent gathering system to do flagging of the correct words. For example, medical intelligent system might need flagging of “illegal drugs” rather than “medical drugs”

   d. **Lexicography**
      WSD and lexicography can work together in loop because modern lexicography is corpus based. With lexicography, WSD provides rough empirical sense groupings as well as statistically significant contextual indicators of sense.

2. **Dictionary Based WSD Approach:**
   As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures and lexical knowledge base. They do not use corpora evidences for disambiguation.
   A major drawback with all of the other approaches of WSD is scalability. All require a considerable amount of work to create a classifier for each ambiguous entry in the lexicon.

   Instead, attempts to perform large-scale disambiguation have focused on the use of machine readable dictionaries.
   In this style of approach, the dictionary provides both the means for constructing a sense tagger, and the target senses to be used.

   The first implementation of this approach is due to Lesk.

In this approach, all the sense definitions of the word to be disambiguated are retrieved from the dictionary. These senses are then compared to the dictionary definitions of all the remaining words in the context. The sense with the highest overlap with these context words is chosen as the correct sense.

The problem with this approach is that dictionary entries for the various senses of target words are relatively short, and may not provide sufficient material to create adequate classifiers. More specifically, the words used in the context and their definitions must have direct overlap with the words contained in the appropriate sense definition in order to be useful.
One way to remedy this problem is to expand the list of words used in the classifier to include words related to, but not contained in their individual sense definitions.

3. **Difficulties in WSD:**
   Followings are some difficulties faced by word sense disambiguation (WSD) −
   a. **Differences between dictionaries**
      The major problem of WSD is to decide the sense of the word because different senses can be very closely related. Even different dictionaries and thesauruses can provide different divisions of words into senses.

   b. **Different algorithms for different applications**
      Another problem of WSD is that completely different algorithm might be needed for different applications. For example, in machine translation, it takes the form of target word selection; and in information retrieval, a sense inventory is not required.

   c. **Inter-judge variance**
      Another problem of WSD is that WSD systems are generally tested by having their results on a task compared against the task of human beings. This is called the problem of interjudge variance.

   d. **Word-sense discreteness**
      Another difficulty in WSD is that words cannot be easily divided into discrete sub meanings.

4. **LESK Algorithm:**

3 WSD using the Lesk Algorithm

The Lesk algorithm [10] uses dictionary definitions (gloss) to disambiguate a polysemous word in a sentence context. The major objective of his idea is to count the number of words that are shared between two glosses. The more overlapping the words, the more related the senses are.

To disambiguate a word, the gloss of each of its senses is compared to the glosses of every other word in a phrase. A word is assigned to the sense whose gloss shares the largest number of words in common with the glosses of the other words. Figure 1 shows the graphic representation of the Lesk Algorithm.

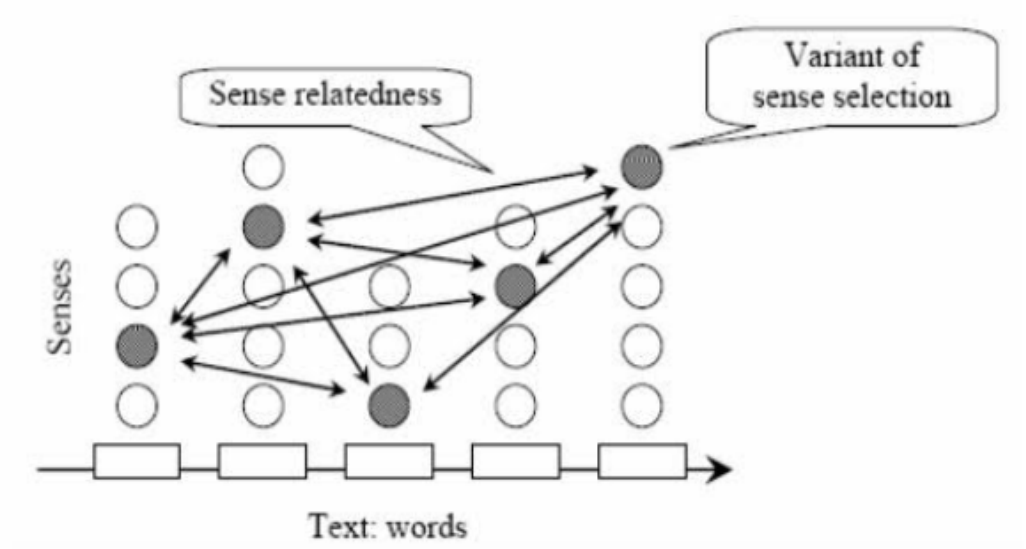


Figure 1. Graphic Representation of the Lesk Algorithm

For example: In performing disambiguation for the "pine cone" phrasal, according to the Oxford Advanced Learner's Dictionary, the word "pine" has two senses:

- sense 1: kind of evergreen tree with needle–shaped leaves,
- sense 2: waste away through sorrow or illness.

The word "cone" has three senses:

- sense 1: solid body which narrows to a point,
- sense 2: something of this shape whether solid or hollow,
- sense 3: fruit of a certain evergreen tree.

By comparing each of the two gloss senses of the word "pine" with each of the three senses of the word "cone", it is found that the words "evergreen tree" occurs in one sense in each of the two words. So these two senses are then declared to be the most appropriate senses when the words "pine" and "cone" are used together.

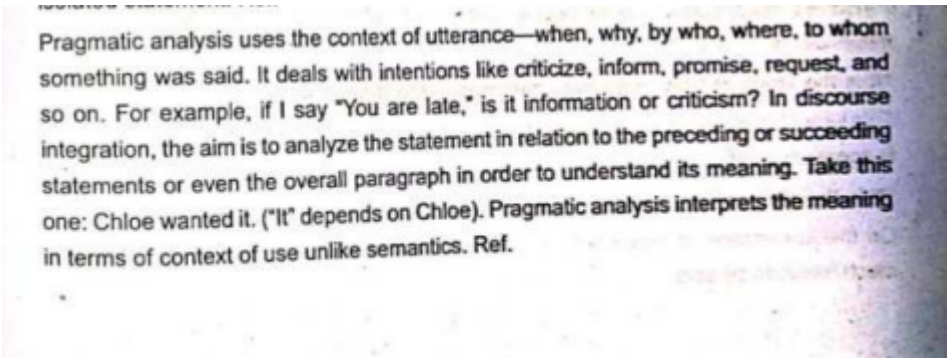
Acording to Padwardhan et al. [14] there are two hypotheses that underly this approach. The first is about the closeness of the words, i.e., the words that appear together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighboring words. The idea behind this hypothesis is, the words that appear together in a sentence generally are related in some way, because to express some idea the human needs a set of words working together. The second hypothesis is that related senses can be identified by finding overlapping words in their definitions. The idea behind this is equally reasonable, in that words that are related will often be defined using the same words, and in fact may refer to each other in their definitions.

The major limitation to this algorithm is that dictionary glosses are often quite brief, and may not include sufficient vocabulary to identify related senses.

```
function Simplified_Lesk(word, sentence)
  best-sense←most frequent sense for word
  max-overlap←0
  context←set of words in sentence
  for each sense in senses of word do
    signature←set of words in the gloss_examples of sense
      overlap←Compute_overlap(signature, context)
      if overlap > max-overlap then
                max-overlap←overlap
                best-sense←sense
  end
  return(best-sense){ returns best sense of word}
```

Module 5:

1. **What is discourse Integration/resolution/Interpretation** actually, its discourse resolution


Pragmatic analysis uses the context of utterance—when, why, by who, where, to whom something was said. It deals with intentions like criticize, inform, promise, request, and so on. For example, if I say "You are late," is it information or criticism? In discourse integration, the aim is to analyze the statement in relation to the preceding or succeeding statements or even the overall paragraph in order to understand its meaning. Take this one: Chloe wanted it. ("It" depends on Chloe). Pragmatic analysis interprets the meaning in terms of context of use unlike semantics. Ref.

Discourse interpretation requires that one build an evolving representation of

discourse state, called a discourse model, that contains representations of the entities that have been referred to and the relationships in which they participate.

Natural languages offer many ways to refer to entities. Each Form of reference sends its own signals to the hearer about how it should be processed with respect to her discourse model and set of beliefs about the world.

Pronominal reference can be used for referents that have an adequate degree of salience in the discourse model. There are a variety of lexical, syntactic, semantic, and discourse factors that appear to affect salience.

Discourses are not arbitrary collections of sentences; they must be coherent. Collections of well-formed and individually interpretable sentences often form incoherent discourses when juxtaposed.

The process of establishing coherence, performed by applying the con-straints imposed by one or more coherence relations, often leads to the inference of additional information left unsaid by the speaker.

The Unsound rule of logical abduction can be used for performing such in-ference.
Discourses, like sentences, have hierarchical structure.
Intermediate Groups of locally coherent utterances are called discourse segments. Discourse structure recognition can be viewed as a by-product of dis-course interpretation.

2. **Explain reference, referent, co-refer, referring expression with examples**
   Reference- the process by which speakers use expressions to denote a person or object.
   Ex:*John went to Bill's car dealership to check out an Acura Integra. He looked at it for about an hour*.
   Words "John" and "he" are used to denote the person named John

   A natural language expression used to perform reference is called a **referring expression**, and the entity that is referred to is called the **referent**.
   Ex: *John went to Bill's car dealership to check out an Acura Integra. He looked at it for about an hour*.
   John and he are referring expressions, and John is their referent.

   Two referring expressions that are used to refer to the same entity are said to **corefer**.
   Ex: *John went to Bill's car dealership to check out an Acura Integra. He looked at it for about an hour*
   John and he corefer in this sentence, as they both refer to John

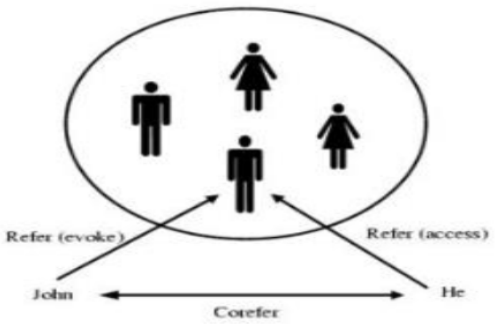
3. **Explain reference resolution and 2 fundamental operations to reference resolution**

18.1 Reference Resolution

- Each type of referring expression encodes different signals about the place that the speaker believes the referent occupies within the hearer's set of beliefs.
- A subset of these beliefs that has a special status from the hearer's mental model of the ongoing discourse, which we call a **discourse model**.
  - The discourse model contains
    - representations of the entities that have been referred to in the discourse and
    - the relationships in which they participate.
- There are two components required by a system to successfully produce and interpret referring expressions:
  - A method for constructing a discourse model that evolves with the dynamically-changing discourse it represents, and
  - A method for mapping between the signals that various referring expression encode and the set of beliefs

18.1 Reference Resolution

- Two fundamental operations to the discourse model
  - **Evoke**
    - When a referent is first mentioned in a discourse, we say that a representation for it is **evoked** into the model.
  - **Access**
    - Upon subsequent mention, this representation is **accessed** from the model.



4. **Types of referent that complicate the reference resolution**

There are three types of referents that complicate the reference resolution problem: inferrables, discontinuous sets, and generics.

In some cases a referring expression does not refer to an entity that has been explicitly evoked in the text, but instead one that is inferentially related to an evoked entity. Such referents are called **inferrables**.
Inferrables can also specify the results of processes described by utterances in a discourse.
Consider the possible follow-ons (a-c) to sentence:

Mix the flour, butter, and water.
a. Kneed the dough until smooth and shiny.
b. Spread the paste over the blueberries.
c. Stir the batter until all lumps are gone.

Any of the expressions the dough(a solid),the batter(a liquid), and the paste(somewhere in between) can be used to refer to the result of the actions

described in the first sentence, but all imply different properties of this result.

Discontinuous sets:
In some cases, references using plural referring expressions like they and them refer to sets of entities that are evoked together, for instance, using another plural expression (their Acuras)or a conjoined noun phrase (John and Mary)

John and Mary love their Acuras. They drive them all the time.

However, plural references may also refer to sets of entities that have been evoked by discontinuous phrases in the text

John has an Acura, and Mary has a Mazda. They drive them all the time.

Here,"they" refers to John and Mary, and likewise "them" refers to the Acura and the Mazda.

Making the reference problem even more complicated is the existence of **generic reference**. Consider example-

I saw no less than 6 Acura Integras today.They are the coolest cars.

Here, the most natural reading is not the one in which "they" refers to the particular 6 Integras mentioned in the first sentence, but instead to the class of Integras in general.

5. **What is anaphora resolution? What is Anaphora? Explain with examples**

Anaphora resolution (AR) which most commonly appears as pronoun resolution is the problem of resolving references to earlier or later items in the discourse. These items are usually noun phrases representing objects in the real world called referents but can also be verb phrases, whole sentences or paragraphs.

Reference to an entity that has been previously introduced into the discourse is called anaphora, and the referring expression used is said to be anaphoric.

There are primarily three types of anaphora:

a. - Pronominal: This is the most common type where a referent is referred by a pronoun. Example: "John found the love of his life" where 'his' refers to 'John'.
b. - Definite noun phrase: The antecedent is referred by a phrase of the form "<the> <noun phrase>". Continued example: "The relationship did not last long", where 'The relationship' refers to 'the love' in the preceding sentence.
c. Quantifier/Ordinal: The anaphor is a quantifier such as 'one' or an ordinal such as 'first'. Continued Example: "He started a new one" where 'one ' refers to 'The relationship' (effectively meaning 'a relationship').

6. **Explain Algorithm for pronoun resolution**

The pronoun resolution algorithm

- Assume that the DM has been updated to reflect the initial salience values for referents.

1. Collect the potential referents (up to four sentences back)
2. Remove potential referents that do not agree in number or gender with the pronouns.
3. Remove potential referents that do not pass intrasentential syntactic coreference constraints.
4. Computed the total salience value of the referent by adding any applicable values from Fig. 18.6 to the existing salience value previously computed during the discourse model update step (i.e., the sum of the applicable values from Fig. 18.5)
5. Select the referent with the highest salience value. In the case of ties, select the closest referent in terms of string position (computed without bias to direction)

(18.68) John saw a beautiful Acura Integra at the dealership.
He showed it to Bob.
He bought it.

We first process the first sentence to collect potential referents and computed their initial salience values.

- No pronouns to be resolved in this sentence.

	Rec	Subj	Exist	Obj	Ind-Obj	Non-Adv	HeadN	Total
John	100	80				50	80	310
Integra	100			50		50	80	280
dealership	100					50	80	230

We move on to the next sentence.

- Gender filtering: he ⇒ John

Referent	Phrases	Value
John	{*John*}	155
Integra	{*a beautiful Acura Integra*}	140
dealership	{*the dealership*}	115

After he is resolved in the second sentence, the DM is updated as below.

- The pronoun in the current sentence (100); Subject position (80); Not in adverbial (50); Not embedded (80)
- Total 310 added to the current weight for John to become 465

Referent	Phrases	Value
John	{*John*, he_1}	465
Integra	{*a beautiful Acura Integra*}	140
dealership	{*the dealership*}	115

- For the next pronoun *it* in the second sentence
  - The referent Integra satisfies parallelism: 140+35= 175
  - The referent dealership: 115
  - ∴ the Integra is taken to be the referent
- Update the DM:
  - *it* receives 100+50+50+80=280
  - Update to become 420
  - Bob= 100+40+50+80=270

Referent	Phrases	Value
John	{*John*, he_1}	465
Integra	{*a beautiful Acura Integra*, it_1}	420
Bob	{*Bob*}	270
dealership	{*the dealership*}	115

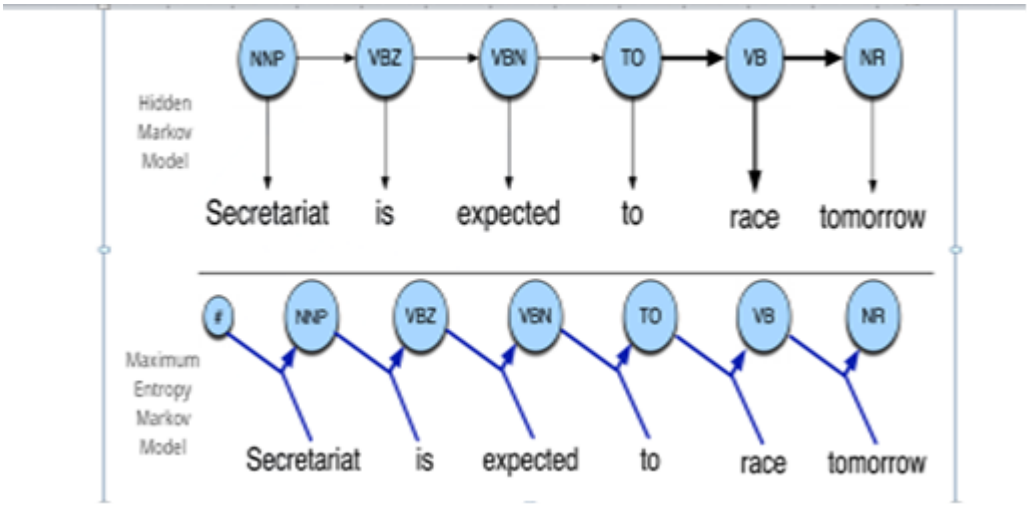
Move on to the next sentence, the DM becomes as follows.

- According to the weights, it is clear to resolve *he* and *it* in the last sentence.

Referent	Phrases	Value
John	{*John*, he_1}	232.5
Integra	{*a beautiful Acura Integra*, it_1}	210
Bob	{*Bob*}	135
dealership	{*the dealership*}	57.5

Assignment 2:

How is MEMM different from Hidden Markov Model (HMM)?



HMM is a **generative model** because words as modelled as observations generated from hidden states. Even the formulation of probabilities uses likelihood P(W|T) and prior P(T). On the other hand, MEMM is a **discriminative model**. This is because it directly uses posterior probability P(T|W); that is, probability of a tag sequence given a word sequence. Thus, it discriminates among the possible tag sequences.

It can also be said that HMM uses **joint probability** for maximising the probability of the word sequence. MEMM uses **conditional probability**, conditioned on previous tag and current word.

In HMM, for the tag sequence decoding problem, probabilities are obtained by training on a text corpus. In MEMM, we build a distribution by adding features, which can be hand crafted or picked out by training. The idea is to select the maximum entropy distribution given the constraints specified by the features. MEMM is more flexible because we can add features such as capitalization, hyphens or word endings, which are hard to consider in HMM. MEMM allows for diverse non-independent features.

HMM - generative model
MEMM - discriminative model

HMM - uses join probability
MEMM - uses discrete probability

HMM - probabilities are obtained by training on a text corpus
MEMM - build a distrbution by adding features, which can be hand crafted or picked out by training

HMM - less flexible (cannot consider capitalization, hyphens or word endings)
MEMM - more flexible (allows diverse non-independent features)

Do we have assignment 2 in question bank?
Yesss