Literature Survey: Bilingual Dependency Parsing and Disambiguation of Prepositional Phrase Attachments

Sagar Sontakke
IIT Bombay
sagarsb@cse.iitb.ac.in

Dependency parsing plays a vital role in major NLP (Natural Language Processing) applications like Machine Translation, Entity Extraction, etc. However, getting a correct dependency parse tree may prove to be a difficult task. There are multiple reasons as follows:

1. Poor Resources: Some languages may not have properly annotated data, which makes it nearly impossible to build or train a highly accurate dependency parser.

2. Lack of NLP Tools: Basic NLP tools which help for dependency parsing may not be available for some languages. These tools include lemmatizer, Part of Speech tagger (POS), Named Entity Recognizer (NER), etc.

To solve the above problem, we can make use of a language which has ample NLP resources and tools. So the biligual approach is useful for the dependency parsing and PP-attachment correction. In the following sections, we look at several bilingual approaches

1 Bilingual Parsing

Parallel treebanks have been receiving interest in recent years, primarily due to their potential use in statistical machine translation. In the following sections, we look into some existing approaches for bilingual parsing and building parallel treebanks.

1.1 Transfer of Delexicalized Parsers

Delexicalized parsers have been used to directly transfer between languages, producing significantly higher accuracies than unsupervised parsers (McDonald et al., 2006). The authors use a constraint driven learning algorithm where constraints are drawn from parallel corpora to project the final parser. They show that simple methods for introducing multiple source languages improve the overall quality of the resulting parsers. They report results in eight languages.

1.2 Joint Parsing and Alignment with Weakly Synchronized Grammars

(Burkett et al., 2010) present a unified joint model for simultaneous parsing and word alignment. To flexibly model syntactic divergence, the authors have developed a discriminative log-linear model over two parse trees and an ITG derivation which is encouraged but not forced to synchronize with the parses.

The model exploits synchronization where possible to perform more accurately on both word alignment and parsing, but also allows independent models to dictate pieces of parse trees and word alignments when synchronization is impossible. This notion of ”weak synchronization” is parameterized and estimated from data to maximize the likelihood of the correct parses and word alignments.

1.3 Bilingual Informed Parsing

(Haulrich, 2012) present three data-driven approaches that exploit bilingual information. Bilingually informed parsing is monolingual parsing that is informed by the syntactic structures of sentences parallel to those being parsed.

Next, they present an iterative approach that rests on the assumption that the better the structures that guide the parsing, the better the output of the parser.

Thirdly, they propose a classic reranking approach where monolingual parses are reranked, based on bilingual features. The authors report considerable improvements over the baseline for the first and the third approaches.

2 Dual Decomposition

Many problems in NLP require an argmax computation of the form $y^* = argmax_{y \in Y} f(y)$, which essentially means finding that y for which the score (is maximized. In the case of parsing, for instance, this stands for finding the score for all possible parse trees $y \in Y$ for a given sentence , and return that which has the highest value for (i.e., the parse tree with the highest score. This is called the decoding problem.

Decoding problems in NLP can be solved using Dual Decomposition. Dual decomposition, or more generally, Lagrangian Relaxation, is a classical method for combinatorial optimization and has been applied to several inference problems in NLP (Rush, 2012). It involves solving of complicated optimization problems by decomposing them into two or more sub-problems, and solving iteratively.

The solutions to the sub-problems have some sort of agreement which is enforced in the form of linear constraints. The chosen sub-problems can be solved efficiently using exact combinatorial algorithms. The agreement constraints are incorporated using Lagrange multipliers, and an iterative algorithm for example, a sub-gradient algorithm is used to minimize the resulting dual.

Dual decomposition algorithms have the following properties:

- They are typically simple and efficient. For example, sub-gradient algorithms involve two steps at each iteration: first, each of the sub-problems is solved using a combinatorial algorithm; second, simple additive updates are made to the Lagrange multipliers.
- They have well-understood formal properties, in particular through connections to linear programming (LP) relaxations.
- In cases where the underlying LP relaxation is tight, they produce an exact solution to the original decoding problem, with a certificate of optimality. In cases where the underlying LP is not tight, heuristic methods can be used to derive a good solution; alternatively, constraints can be added incrementally until the relaxation is tight, at which point an exact solution is recovered.

The agreement constraints are incorporated using Lagrange multipliers, and an iterative algorithm for example, a sub-gradient algorithm is used to minimize the resulting dual. To understand how dual decomposition works, let us take an example from (Rush and Collins, 2012).

Consider the problem of finding the constituency parse tree of a sentence and the parts-of-speech (POS) tags of the words in the sentence (Figure 1 and Figure 2).

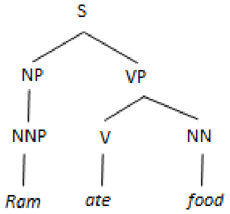


Figure 1: Constituency Parse Tree

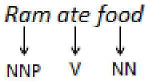


Figure 2: POS Tagged Sentence

For a given sentence of n words, we maximize:

$$argmax_{y \in Y, z \in Z} f(y) + g(z)$$

such that:

$$\forall i \in \{1 \; to \; n\} \;\; \forall t \in T, y(i,t) = z(i,t)$$

Adding the constraints to the objective function by introducing Lagrangian multipliers u

$$L(u,y,z) = f(y)+g(z)+\sum_{i,t} u(i,t)(y(i,t)-z(i,t))$$

By distributing terms and rearranging we obtain the following equation

$$L(u,y,z) = (f(y) + \sum_{i,t} u(i,t)y(i,t))$$

$$+(g(z) - \sum_{i,t} u(i,t)z(i,t))$$

The Lagrangian dual of the problem becomes:

$$L(u) = argmax_{y \in Y, z \in Z} L(u, y, z)$$

$$= argmax_{y \in Y}(f(y) + \sum_{i,t} u(i,t)y(i,t))$$

Our objective is to find the best constituency parse and POS tags for a given sentence, such that they agree on the tag labels. Let $f(y)$ be the score of the Constituency Parser and $g(z)$ be the score of the POS tagger. Let T be the set of all POS tags. For any parse tree y, for any position $i \in 1ton$, for any tag $t \in T$, $y(i,t) = 1$ if parse tree y has tag t at position i and $y(i,t) = 0$ otherwise. For a tag sequence, $z(i,t) = 1$ if the tag sequence has tag i at position i, 0 otherwise.

Assuming that the individual maximization problems can be solved efficiently, we can solve the above optimization problem by iteratively updating the sub-gradient.

3 Joint Inference of NLP Tasks

Problems at the higher of the NLP pyramid often need to be solved by the aid of the lower level NLP tasks. POS tagging, for example, is essential for the task of syntactic parsing. Similarly, in the tasks of morphology analysis and POS tagging, POS tagging may use word morphology as a feature, while a morphology analyzer may also use POS tag information for disambiguation in obtaining the correct word roots and suffixes. Joint inference is an effective approach to avoid cascading of errors when inferring multiple natural language tasks. It allows bidirectional flow of information, allowing for corrections to be made for tasks earlier in the pipeline using the output of later tasks.

Also, there are multiple NLP tasks related to one another, and it makes sense to take a joint approach in solving them. The tasks of word sense disambiguation and semantic role labelling draw heavily from one another.

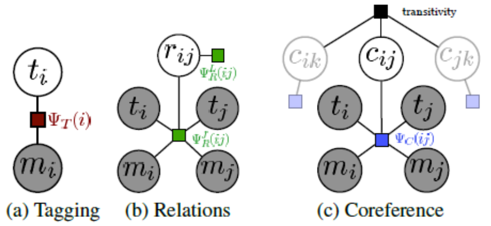
Joint inferencing has an important role to play in the situation of performing the same NLP task across languages. In these scenarios, the power of jointly performing the task comes from parallel corpora. It has been shown that bilingual texts annotated with NER tags can provide useful additional training sources for improving the performance of standalone monolingual taggers (Wang et al., 2013). This is because text in two languages may contain complementary cues that help to disambiguate named entity mentions.

In the current work, we jointly infer dependency parsing for a pair of language, and also, use joint inference for the tasks of dependency parsing and word alignments.

3.1 Joint Modelling of Entities, Relation Extraction and Coreference Resolution

(Singh et al., 2013) have proposed a single, joint probabilistic graphical model for classification of entity mentions (entity tagging), clustering of mentions that refer to the same entity (coreference resolution), and identification of the relations between these entities (relation extraction).

Entity tagging is the task of classifying each entity mention according to the type of entity to which they refer. The input for this task is the set mention boundaries and the sentences of a document. For each mention m_i, the output of entity tagging is a label t_i from a predefined set of labels T. The set of labels used in newswire consist of PERSON, ORGANIZATION, GEO-POLITICAL, LOCATION, FACILITY, VEHICLE, and WEAPON. It is generally modelled as a Maximum Entropy Model. This model may be written as a graphical model by defining a factor $\psi_T(m_i, t_i)$,for each entity tag variable t_i.



(a) Tagging (b) Relations (c) Coreference

Figure 3: Individual Classification Models

Relation extraction labels each entity mention pair in the same sentence with its relation as expressed in that sentence, or NONE if no relation is expressed. This task is often represented as variables $r_{i,j}$ that represent the type of the relation

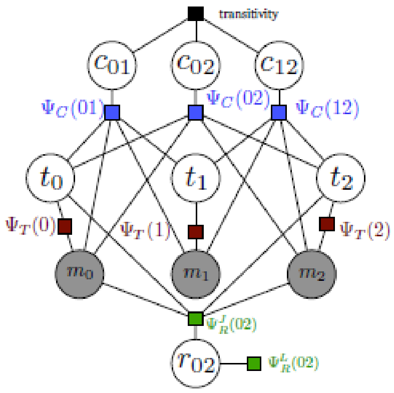


Figure 4: Joint Model of Entity Tagging, Resolution and Relations

where m_i, is the first argument, m_j, the second argument, and the type comes from a predefined set of labels R^3. The model is represented as factor templates $\psi_R^L(m_i, m_j, r_{ij})$ and $\psi_R^L(m_i, m_j, t_i, t_j)$ over variables.

Coreference is the task of linking mentions within a document that refer to the same real-word entity. Given the mentions in a document, and the coreference system predicts entities by identifying links between the mentions. A common approach to the coreference task is to classify pairs of mentions as coreferent or not, i.e. for pairs of mentions m_i and m_j that appear in the same document, there is a variable $c_{ij} \in 0, 1$. The parameters of the model are defined by a factor template $\psi_C(C_{ij}, m_i, m_j, t_i, t_j)$ as shown in Figure 3

A model is defined that directly represents the dependencies between the three tasks by modeling the joint distribution over the three tasks (Figure 4).

$$p(t, r, c|m) \propto \left(\prod_{t_i \in T} \psi_T(m_i, t_i) \right.$$

$$\cdot \prod_{c_{i,j} \in C} \psi_C(C_{i,j}, m_i, m_j, t_i, t_j)$$

$$\left. \cdot \prod_{r_{i,j} \in r} \psi_R^L(m_i, m_j, r_{i,j}) \psi_R^L(m_i, m_j, t_i, t_j) \right)$$

Instead of representing a distribution over the labels of a single task conditioned on the predictions from another task, these factors now directly represent the joint (un-normalized) distribution over the tasks that they are defined over.

3.2 Joint inference of NER and Alignment

(Wang et al., 2013) developed a bilingual NER model by embedding two monolingual CRF-based NER models into a large undirected graphical model, and introducing additional edge factors based on word alignment. They also propose an extension with two uni-directional HMM-based alignment models, and perform joint decoding of NER and word alignments.

The new model factors over one NER model and one word alignment model for each language, plus a joint NER-alignment model which not only enforces NER label agreements but also facilitates message passing among the other four components.

Let us take a closer look at their formulation. Let $k(y^e)$ be the un-normalized log-probability of tag sequence y^e.

At inference time, the maximization objective is:

$$max_{y^e, y^f} k(y^e) + l(y^f)$$

$$\text{such that } \forall (i, j) \in A, y_i^e = y_j^f.$$

i.e. for parallel sentence in two languages, for every pair of aligned words, the NE tag should be same for both languages. A dual decomposition based inference algorithm has been used for decoding.

4 Disambiguation of Prepositional Phrase Attachments

4.1 Need for PP-attachment Disambiguation

Prepositional phrase (PP) attachment is a major source of ambiguity in languages like English. It has a substantial challenge to Machine Translation (MT) between English and languages that are not characterized by PP attachment ambiguity. English is syntactically ambiguous with respect to PP attachment.

Consider this example:

I washed a jeans with pockets.

The attachment for the preposition *with* is ambiguous. Syntactically, it can attach to the verb *wash* or with the noun *jeans*. This a problem of PP attachment disambiguation and needs to be solved for applications like Machine Translation.

4.2 Current State of the Art

PP Attachment Disambiguation using Multilingual Aligned Data

The work by Lee Schwartz and Takako Aikawa (Schwartz et al., 2003) focuses on solving the English PP attachment problem with the help of multilingual aligned parallel data with an unsupervised system. They have used English-Japanese parallel data to solve English PP attachment disambiguation problem.

The approach is unsupervised, but it does require a large, parsed, sentence-aligned, bilingual corpus. It exploits the unambiguous nature of PP attachment in Japanese. In this system, reattachment of English PPs takes place in the English analysis component after an initial parse is produced. By design, the initial parse has low right attachments of PPs. The reattachment module traverses the nodes of the parse tree and marks all the potential attachment sites for each PP. two different types of data: (i) data that serve as positive evidence for VP attachment (ii) data that serve as negative evidence for VP attachment. Positive evidence consists of examples for which VP attachment is suggested by the Japanese data. Negative evidence consists of examples for which NP attachment is suggested.

A Rule-Based Approach to PP Attachment Disambiguation

This work by (Brill and Resnik, 1994) is a rule-based approach to prepositional phrase attachment, disambiguation. A set of simple rules is learned automatically to try to prediet proper attachment based on a number of possible contextual cues. It employs a "Transformation-Based Error-Driven Learning System" for PP attachment.

Figure 5 shows the steps. First, unannotated text is passed through the initial annotator. The annotated text is then compared with the truth i.e. the manually annotated data and transformations are learned that are applied to the output of the initial state annotator to make it better resemble the truth.

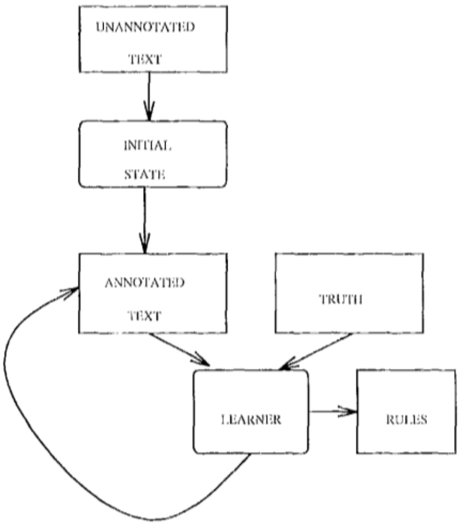


Figure 5: Transformation-Based Error-Driven Learning

Semantic Dictionary for PP Attachment Disambiguation

(Stetina and Nagao, 1997) proposed a supervised learning method for PP attachment based on a semantically tagged corpus. Many a times, the PP attachment is based on the contextual information. But do not have a computer database containing life time experiences, and therefore we have to find another way of how to decide the correct PP attachment. One of the solutions lies in the exploration of huge textual corpora, which can partially substitute world knowledge.

A number of supervised and unsupervised approaches for solving the PP-attachment problem have been proposed in the literature. (Ratnaparkhi et al., 1994) uses a Maximum Entropy Model for solving the PP-attachment decision. (Agirre et al., 2008) have used WSD-based strategies in different capacities to solve the problem of PP-attachment. (Olteanu and Moldovan, 2005) have attempted to solve the PP-attachment problem as a classification problem of attachment either to the preceding verb or the noun, and have used Support Vector Machines (SVMs) that use complex syntactic and semantic features.

5 Statistical Machine Translation

5.1 Alignment

Alignment is the mapping between the words in one language to another. An alignment function maps each output word in the foreign language at position j to an English input word at position i.

Let us assume that the task is to translate from a foreign language F (the ”source” language) into English (the ”target” language) Let F be a sentence in the foreign language with word $f = (f_1, f_2, \ldots, f_{l_f})$, where l_f is the length of the sentence.
and f_j for j ∈ $\{1tom\}$, is the j^{th} word in the sentence. e refers to an English sentence, e is equal to $e = (e_1, e_2, \ldots, e_{l_e})$ where l_e is the length of the English sentence.

The alignment function maps each English output word at position i to a German input word at position to j

$$a : i \rightarrow j \tag{1}$$

5.2 IBM Models

Models that decompose a joint probability P(x, y) into terms P(x) and P(x|y) are known as noisy channel models. The IBM models (**?**) are an instance of a noisy-channel model, and they have two components:

- A language model that assigns a probability p(e) for any sentence $e = (e_1,\ e_2,\ \ldots,\ e_l)$ in English. The parameters of the language model are estimated from very large quantities of English data.

- A translation model that assigns a conditional probability p(f|e) to any foreign/English pair of sentences. The parameters of this model are estimated from the translation examples.

Given these two components of the model, following the noisy channel approach, the output of the translation model on a new foreign language sentence f is:

$$e^* = argmax_{e \in E} p(e) \times p(f|e) \tag{2}$$

where E is the set of all sentences in English. Thus the score for a potential translation is the product of two scores: first, the language-model score p(e) , which gives a prior distribution over which sentences are likely in English; second, the translation-model score p(f|e), which indicates how likely we are to see the French sentence f as a translation of e.

Five models of increasing complexity were proposed in the original work on statistical machine translation at IBM. The advances of the five IBM models are: IBM Model 1: lexical translation; IBM Model 2: adds absolute alignment model; IBM Model 3: adds fertility model; IBM Model 4: adds relative alignment model; IBM Model 5: fixes deficiency.

In the following sections, we look at these models briefly.

IBM Model 1

IBM Model 1, which is a generative model for sentence translation, is based solely on *lexical translation probability* distributions. Lexical translation probability is simply based on the count of the times a word in the foreign language is translated to a word in English in a large parallel corpus. For each output word e that is produced by our model from an input word f , we want to factor in the translation probability p(f|e), and nothing else.

We define the translation probability for a foreign sentence $f = (f_1, f_2, \ldots, f_{l_f})$ of length l_f to an English sentence $e = (e_1, e_2, \ldots, e_{l_e})$ of length l_e with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \rightarrow i$ as follows:

$$p(e, a|f) = \frac{\epsilon}{(l_f + 1)^l_e} \sum_{j=1}^{l_e} t(e_j | f_{a(j)}) \tag{3}$$

Let us take a closer look at this formula. The core is a product over the lexical translation probabilities for all l_e generated output words e_j. The fraction before the product is necessary for normalization. Since we include the special NULL token, there are actually $l_f + 1$ input words. Hence, there are $(l_f + 1)^{l_e}$ different alignments that map $l_f + 1$ input words into l_e output words. The parameter ϵ is a normalization constant, so that $p(e, a|f)$ is a proper probability distribution, meaning that the probabilities of all possible English translations e and alignments a sum up to one: $\sum_{e,a} p(e, a|f) = 1$

IBM Model 2

IBM Model 2 adds an explicit model for alignment. The translation of a foreign input word in position *i* to an English word in position *j* is modeled by an alignment probability distribution $a(i|j, l_e, l_f)$

IBM Model 2 is a two-step process, with a lexical translation step and an alignment step. The first step is lexical translation as in IBM Model 1, again modeled by the translation probability t(e | f). The second step is the alignment step. The two steps are combined mathematically to form IBM Model 2:

$$p(e, a|f) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) a(a(j)|j, l_e, l_f) \quad (4)$$

IBM Model 3

IBM model 3 also models the *fertility* of output words. Fertility is the notion that input words produce a specific number of output words in the output language. A word in the source language may correspond to multiple words in the target language.

The fertility of input words is modelled with a probability distribution $n(\varphi|f)$

For each foreign word *f*, this probability distribution indicates how many φ = 0, 1, 2, ... output words it usually translates to. The parameters of Model 3 are a set of fertility probabilities, translation probabilities and distortion probabilities, which gives us

$$p(e|f) = \sum_a p(e, a|f)$$

$$= \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_f} \binom{l_e - \varphi_0}{\varphi_0} p_1^{\varphi_0} p_0^{l_e-\varphi_0} \prod_{i=1}^{l_f} \varphi_i! n(\varphi_i|f_i) \quad (5)$$

$$\times \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) d(j|a(j), l_e, l_f) \quad (6)$$

IBM Models 4 and 5

In IBM Model 4, a relative distortion model is introduced. In this *relative distortion model*, the placement of the translation of an input word is typically based on the placement of the translation of the proceeding input word. A problem with Model 3 and Model 4 is that in these models, it is possible that multiple output words may be placed in the same position. This is called deficiency. The distortion model in IBM Model 5 handles *deficiency* and is based on vacant word positions.

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and pp attachment performance with sense information. In *ACL*, pages 317–325. Citeseer.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 1198–1204. Association for Computational Linguistics.

David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–135. Association for Computational Linguistics.

Martin Wittorff Haulrich. 2012. *Data-driven bitext dependency parsing and alignment*. Copenhagen Business SchoolCopenhagen Business School, Department of International Business Communication-Department of International Business Communication.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 216–220. Association for Computational Linguistics.

Marian Olteanu and Dan Moldovan. 2005. Pp-attachment disambiguation using large context. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 273–280. Association for Computational Linguistics.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the workshop on Human Language Technology*, pages 250–255. Association for Computational Linguistics.

Alexander M Rush and Michael Collins. 2012. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. *Journal of Artificial Intelligence Research*.

Lee Schwartz, Takako Aikawa, and Chris Quirk. 2003. Disambiguation of english pp attachment using multilingual aligned data. In *Proceedings of MT Summit IX*. Citeseer.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 1–6. ACM.

Jiri Stetina and Makoto Nagao. 1997. Corpus based pp attachment ambiguity resolution with a semantic dictionary. In *Proceedings of the fifth workshop on very large corpora*. Citeseer.

Mengqiu Wang, Wanxiang Che, and Christopher D Manning. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *ACL (1)*, pages 1073–1082.