Abstract: In the last 12 years, there has been a big progress in the field of unsupervised dependency parsing. Different approaches however sometimes differ in motivation and definition of the problem. Some of them allow using resources that are forbidden by others, since they are treated as a kind of supervision. The goal of this paper is to define all the variants of unsupervised dependency parsing problem and show their motivation, progress, and the best results. We also discuss the usefulness of the unsupervised parsing generally, both for the formal linguistics and for the applications.

1 Introduction

Dependency parsing is one of the traditional tasks in natural language processing. It gets a tokenized sentence as input (in most cases, individual tokens (words) are labelled by part-of-speech (POS) tags), and produces a rooted dependency tree, in which the nodes correspond to words and edges correspond to syntactic relations between the words.

Rule-based approaches of dependency parsing were suppressed by the statistical dependency parsers, which achieved better quality compared to the human annotations. Important milestones in dependency parsing were the CoNLL shared tasks in 2006 [4] and 2007 [22]. They provided about 20 treebanks of different languages available in the same format. This became the standard for measuring quality of dependency parsers in fact up to now.

At the same time, there were efforts to develop a parser that does not need any annotated data. The unsupervised parsers infer the dependency structures based on language- and tagset-independent properties of dependency trees, which is mainly the low entropies of the governing-dependent word pairs and low entropies of the word fertilities (number of dependents).

One general motivation is to be able to parse languages for which no annotated treebanks exist. Less sound motivation is to create a dependency structure which better suits a particular NLP application, e.g. machine translation.

This is a survey paper about unsupervised dependency parsers. Since different approaches have different motivations, allow to use different kinds of data and different amount of knowledge about them, they cannot be compared because of different degree of (un)supervision. The aim of this paper is to cluster the approaches to several groups in which they are comparable and to show the most important ones together with the results.

The paper is structured as follows: In Section 2 we define different unsupervised parsing problem settings and summarize the motivations and advantages. Section 3 describes different evaluation measures developed for unsupervised parsers. In Section 4 we go through the works done in this field and describe the most important approaches. In Section 5 we compare the results across data, parsers, and languages. Section 6 discusses generally the usefulness of unsupervised parsing methods in linguistics and in applications. Section 7 concludes.

2 Problem Settings

Some unsupervised parsing approaches use different kinds of data for the grammar inference than others. What is used in one is treated as not allowed kind of supervision in another. We therefore categorize the approaches into four groups. They are described in the following subsections and sorted from the least unsupervised (more data and knowledge) to the most unsupervised (less data and knowledge).

2.1 Using supervised POS tags and some knowledge about them

In the first group of approaches, there are parsers that need the sentences labelled by supervised POS tags, i.e. by a manually designed tagset. On top of that they also somehow utilizes the knowledge about the tagset. For example, they know which tags are used for verbs and therefore treat them differently through the grammar inference. This is the main difference from the second group (Section 2.2) and it is sometimes considered as a bit of cheating. If we know the meaning of the POS tags, we could easily build a simple rule-based parser, which would definitely not be unsupervised. This also relates to so-called delexicalized parsing, where the parser is trained on a different language with the same POS tagset and the model can then be used for languages without treebanks. This is however beyond the scope of this paper. The approaches we assigned to this group however use only a bit of such knowledge that help the inferred structures to be in a required shape.

In recent years, many researchers work on a project called Universal Dependencies [21], a collection of treebanks for many languages (51 treebanks and 40 languages in its current version 1.5), where the morphological and dependency annotation styles are unified across the languages.
2.2 Using supervised POS tags without any knowledge about them

The majority of works describing unsupervised dependency parsing utilize supervised POS tags without any knowledge about them. In other words, the parsers take the POS tags only as labels without any meaning. It is a bit strange not to tell the parser anything, for example: “ADJ are adjectives and often depends on the following NOUNs”, if there is such possibility, however allowing it would bring the parsers to the first group (Section 2.1) whose unsupervisedness is sometimes disputable.

Nevertheless, what is more strange, is the usage of supervised POS tags. The POS tags carry a lot of syntactic information. Imagine a sequence of POS tags “ADJ NOUN VERB PREP ADJ NOUN”. You would easily build the most probable dependency tree. The motivation of this problem setting may be:

1. We want to compare supervised and unsupervised parsers operating on the same tagset.

2. We want to evaluate an unsupervised parser and, in the future, we will use the unsupervised word classes instead of the supervised tags on low-resourced languages and hope that it will work as well.

3. We have a language without treebank and we have a POS tagger. However, we are not able to find the meaning of POS tags used.

The third option is rather hypothetical. We always find someone who speaks that language or have a parallel corpus from which we could get basic meanings of individual words and tags.

It is also worth to mention that almost all the experiments and evaluation in the papers were done using gold standard POS tags, i.e. the POS tags assigned manually by human annotations. This is not surprising. While the qualities of unsupervised parsers are substantially lower than the qualities of supervised parsers, it is not worthy to make experiments also with the predicted POS tags.

2.3 Using unsupervised POS tags

The lower attention was given to fully unsupervised parsers using unsupervised POS tags (word classes). The only source they use is raw text. The motivation is obvious here: If we want to analyze a language without any manually annotated resources, we need exactly this approach. Other motivation could be the need of having different structures from that present in annotated treebanks. Majority of works here used the same parser as for supervised POS tags (Section 2.1) and obtain the unsupervised POS tags by some of the best word clustering tools available.

2.4 Direct parsing from raw text without POS tags

The last setting we describe is unsupervised parsing from raw texts. Here we do not use any POS tags or word classes. The only units the parser plays with are the words. The results should be theoretically compared with the previous category (2.3), where unsupervised word clustering is used. However, the word classes are typically inferred on much larger text corpora than dependency trees are. This approaches use therefore much less data for the inference and that is why we assign them into a separate category. Such approaches would be the most elegant way of parsing, however, they naturally achieve very poor results.

3 Parsing Evaluation

The unsupervised parsing approaches sometimes differ also in evaluation metrics. The standard attachment score is sometimes found too strict to evaluate the inferred structures and therefore new, more tolerant metrics, are designed. The following three evaluation metrics exists:

1. Directed attachment score (unlabeled attachment score) is a standard metric for measuring dependency parsing quality. It is a percentage of words correctly attached to their parents. It does not allow even the slightest local structural differences, which might be caused just by more or less arbitrary linguistic or technical conventions.

2. Undirected attachment score disregards the directions of edges and is therefore less biased towards such conventions. For example, there is no difference whether the parser attaches prepositions to nouns or nouns to prepositions. Nevertheless, this holds for all edges, including these with undoubted directions.

3. Neutral edge direction metric proposed by [24] is even more tolerant in assessing parsing errors than the undirected attachment score. It treats not only node’s parent and child as the correct answer, but also its grandparent.

Even though the alternative scores were proposed and sometimes used, the majority of experiments were evaluated by the directed attachment scores, probably because of its simplicity and the tradition in the field and also because the other two did not prove to be substantially better.

4 Unsupervised Dependency Parsers

In this Section, we summarize and describe the most important works in the field of unsupervised dependency

---

2 We do not want to use the abbreviation UAS for the unlabeled attachment score here, since it could be mistaken for undirected attachment score.

3 [http://www.cs.huji.ac.il/~roys02/software/ned.html](http://www.cs.huji.ac.il/~roys02/software/ned.html)
parsing through the last 12 years. Even though there were a couple of works before, the first paper with results better than a chain baseline was the Dependency Model with Valence by Klein and Manning [13].

We first describe the methods using supervised POS tags without any other knowledge (Section 2.2) in Sections 4.1 and 4.2, then we switch to other settings. A detailed table with results over different methods and different problem settings is shown in Section 5.

4.1 Dependency Model with Valence

We start with Dependency Model with Valence (DMV), which was introduced by Klein and Manning [13]. It is the most popular approach, which was followed by many other researchers and improved in many ways. It is a generative model that generates dependency trees using two submodels:

- Stop model $p_{stop}(|t_{g}, d|)$ represents probability of not generating another dependent in direction $d$ to a node with POS tag $t_{g}$. The direction $d$ can be left or right. If $p_{stop} = 1$, the node with the tag $t_{g}$ cannot have any dependent in direction $d$. If it is 1 in both directions, the node is a leaf.

- Attach model $p_{attach}(|t_{d}|t_{g}, d|)$ represents probability that the dependent of the node with POS tag $t_{g}$ in direction $d$ is labeled with POS tag $t_{d}$.

The grammar consisting of probability distributions $p_{stop}$ and $p_{attach}$ is learned using the Expectation Maximization inside-outside algorithm [12]. The learning is further improved by Smith et al. [26] and Cohen et al. [8]. Headen et al. [11] introduce the Extended Valence Grammar and add lexicalization and smoothing. Besides the POS tags, the parser begin to operate with word forms as well. Blunsom and Cohn [2] use tree substitution grammars, which allow learning of larger dependency fragments by employing the Pitman-Yor process. Spitkovsky [30] improves the inference using iterated learning of increasingly longer sentences. Further improvements are achieved by better dealing with punctuation [32] and new “boundary” models [33]. Spitkovsky also improves the learning itself in [31] and [34].

Mareček and Straka [16] use so called reducibility principle to predict $p_{stop}$ probabilities for individual POS tags from raw texts, add it to the Dependency Model with Valence and use Gibbs sampling to infer the grammar. In [19], they suppose that the function words, which can be predicted by their shortness, have fixed low number of dependents and move the parsing results even a bit higher.

4.2 Other approaches using supervised POS tags

There are also approaches not based on DMV, even though their models are not far from it. Mareček and Žabokrtský [18] use a fertility to model number of children for particular POS tags instead of the $p_{stop}$ model.

Sogaard [27] explores a completely different view in which a dependency structure is among other things a partial order on the nodes in terms of centrality or saliency. Cohen et al. [7] do the grammar inference multilingually on more languages. The data do not need to be parallel, they only have to share the tagset. The inference is then less prone to skew to bad solutions due to the language differences.

Bisk and Hockenmaier [1] use the Combinatory Categorial Grammars for dependency structure induction.

4.3 Approaches using some knowledge about the POS tags

The “less unsupervised” approaches utilizing an external knowledge of the POS tagset reach often better attachment scores than the previous approaches. Any additional knowledge about the tags used can be very strong and can change the inferred structures dramatically. For example, Naseem et al. [20] follow Eisner [9] and make use of manually-specified universal dependency rules such as $\text{Verb} \rightarrow \text{Noun or Noun} \rightarrow \text{Adjective}$ to guide grammar induction and improve the results by a wide margin. Mareček and Žabokrtský [17] show that only the information that “the POS tags for nouns are more frequent than the POS tags for verbs” very much improves the baseline. This however fails for example in case the POS tags for nouns are subcategorized in some way. Then we would need to know which POS tags are for nouns and group them together. Rasooli and Faiili [23] identify the last verb in the sentence, minimize its probability of reduction and push it to the root position, and also make a huge improvement.

Such approaches achieve better results; however, they are useless for grammar induction for languages, for which the tagger is not available.

4.4 Approaches using unsupervised POS tags

These approaches mostly do not bring any new methods. The authors only take their unsupervised parsers we presented in Section 4.1 and take a word clustering tool to produce unsupervised POS tags and run their parser on them. Spitkovsky [29] took the clustering tool by Clark [6] and Brown et al. [3] and showed that the parsing with supervised POS tags can be outperformed for English, if the word classes are used instead. Mareček [15] performed similar experiments on 30 languages and showed that on some of them the use of unsupervised word classes instead of supervised POS tags improve the parsing accuracy. The average score across the languages was however significantly worse.
Christodoulopoulos et al. [5] try to do inference of POS tags and dependency structure together. After random initialization, they alternate the prediction of the structure based on the POS tags and prediction of the POS tags based on the structure.

4.5 Approaches using raw text only

There are couple of approaches, which do not need any word categorization. We only mention the incremental parsing by Yoav Seginer [25]. His algorithm collects lists of labels for each word, based on neighboring words, and then directly uses these labels to parse.

5 Results

In Tables 1, 2, and 3 we summarize the results over the individual parsers, data, and settings. Unfortunately, different parsers were evaluated on different data. In the beginnings, the parsers were evaluated mainly on the English Penn Treebank [14] (transformed to dependencies) and some only on the short sentences of length up to 10 (ptb10), since the shorter sentences were easier to parse and the resulting scores did not look so bad. See Table 1.

After the unsupervised parsers were improved and achieved much better results than simple baselines, they started to be evaluated across languages and on sentences of all lengths (Table 2).

In 2012, there has been a shared task on unsupervised dependency parsing named “The PASCAL challenge on Grammar Induction” [10]. Seven competing parsers were evaluated on new datasets comprising ten different languages, including simpler English used by small children. See Table 3.

Unfortunately, some of the parsers were evaluated on non-standard data or with non-standard metrics and therefore their results could not be added into any of the three tables.

All the tables share the same format: each method is labelled by a link to the references and by a group label: SP for using supervised POS tags, UP for using unsupervised POS tags, and SP+K when an additional knowledge about the supervised tags was used.

6 Usefulness of Unsupervised parsers in linguistics and applications

We could see a lot of work done in the field of unsupervised parsing in the last 12 years. The quality of induced structures are better than before, but the supervised parsers are still better than the unsupervised ones by a wide margin. However, for low resourced languages, for which no annotated data exists, this is the way, how to obtain their syntactic structure.

A more serious problem with unsupervised parsing is that, according to our knowledge, there were so far no works incorporating any kind of unsupervised parsing into applications, even though many papers mention that in some cases, an unsupervised structures, different from manual annotations following a given schema, may be very beneficial.

Moreover, in the last two years, no new strong paper about unsupervised parsing appeared in NLP conferences. Instead, a new techniques have arrived: The recurrent neural networks, which may fulfill the previous motivations for unsupervised parsing – to find a structure of language that would help machines to understand it better. Instead of dependency trees, some structures are hidden in hidden states of the deep neural networks.

From the linguistic point of view, the structures inferred by unsupervised parsers can be compared to the manually annotated treebanks. What are the differences? How the unsupervised methods deal with phenomena that are not clear how to parse? Should prepositions depend on nouns or vice versa? And what about coordinations? Many such questions could be answered, however, neither this topic was studied so far.

7 Conclusions

We categorized the unsupervised dependency parsers into four groups according to their needs of data, so that they could be fairly compared. We make a survey over the most important papers and works that reached state-of-the-art results when they were published. We showed a comparison of the results across the methods and languages. It is apparent that there is a big variance over the attachment scores for individual languages. The good performance of a method on one language tells nothing about the performance on another language. We hope that this paper brings to readers some system in the world of unsupervised parsing.

Acknowledgments

This work has been supported by the grant 14-06548P of the Czech Science Foundation.

References


<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>group</td>
<td>–</td>
<td>–</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>UP</td>
</tr>
<tr>
<td>PennTreebank (up to 10 words)</td>
<td>27.7</td>
<td>38.4</td>
<td>47.5</td>
<td>59.3</td>
<td>68.8</td>
<td>67.7</td>
<td>57.1</td>
<td>69.5</td>
<td></td>
</tr>
<tr>
<td>PennTreebank (all sentences)</td>
<td>24.0</td>
<td>31.7</td>
<td>39.0</td>
<td>55.7</td>
<td>45.0</td>
<td>50.8</td>
<td>58.4</td>
<td>59.1</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Results comparison on Penn treebank [14] (automatically converted to dependencies by PennConverter).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>–</td>
<td>–</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>group</td>
<td>–</td>
<td>–</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP</td>
<td>SP+K</td>
<td>UP</td>
<td>SP+K</td>
</tr>
<tr>
<td>Arabic 2006</td>
<td>55.6</td>
<td>7.9</td>
<td>32.6</td>
<td>10.9</td>
<td>9.3</td>
<td>26.5</td>
<td>38.2</td>
<td>35.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arabic 2007</td>
<td>59.0</td>
<td>6.0</td>
<td>35.3</td>
<td>44.9</td>
<td>26.8</td>
<td>27.9</td>
<td>35.3</td>
<td>52.1</td>
<td>24.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basque 2007</td>
<td>23.0</td>
<td>30.5</td>
<td>29.9</td>
<td>33.3</td>
<td>24.4</td>
<td>26.8</td>
<td>35.5</td>
<td>35.3</td>
<td>28.7</td>
<td>34.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bulgarian 2006</td>
<td>38.8</td>
<td>17.9</td>
<td>40.5</td>
<td>65.2</td>
<td>63.4</td>
<td>46.0</td>
<td>54.9</td>
<td>56.8</td>
<td>38.6</td>
<td>53.9</td>
<td>47.9</td>
<td>51.4</td>
</tr>
<tr>
<td>Catalan 2007</td>
<td>30.0</td>
<td>24.8</td>
<td>55.2</td>
<td>62.1</td>
<td>68.0</td>
<td>47.0</td>
<td>67.0</td>
<td>43.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese 2006</td>
<td>14.0</td>
<td>40.7</td>
<td>63.2</td>
<td>58.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese 2007</td>
<td>13.4</td>
<td>41.3</td>
<td>57.0</td>
<td>52.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Czech 2006</td>
<td>26.9</td>
<td>25.2</td>
<td>37.8</td>
<td>55.1</td>
<td>44.0</td>
<td>49.5</td>
<td>52.4</td>
<td>55.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Czech 2007</td>
<td>29.6</td>
<td>24.2</td>
<td>36.2</td>
<td>54.2</td>
<td>34.3</td>
<td>48.0</td>
<td>51.9</td>
<td>54.3</td>
<td>42.4</td>
<td>38.0</td>
<td>33.3</td>
<td></td>
</tr>
<tr>
<td>Danish 2006</td>
<td>47.8</td>
<td>13.1</td>
<td>37.1</td>
<td>22.2</td>
<td>21.4</td>
<td>38.6</td>
<td>41.6</td>
<td>45.5</td>
<td>39.9</td>
<td>53.1</td>
<td>40.8</td>
<td>38.6</td>
</tr>
<tr>
<td>Dutch 2006</td>
<td>24.5</td>
<td>28.0</td>
<td>14.0</td>
<td>46.6</td>
<td>48.0</td>
<td>44.2</td>
<td>47.5</td>
<td>54.8</td>
<td>50.2</td>
<td>48.8</td>
<td>34.4</td>
<td>43.4</td>
</tr>
<tr>
<td>English 2007</td>
<td>21.0</td>
<td>29.4</td>
<td>50.3</td>
<td>29.6</td>
<td>58.2</td>
<td>49.2</td>
<td>55.4</td>
<td>58.5</td>
<td></td>
<td>47.9</td>
<td>23.8</td>
<td></td>
</tr>
<tr>
<td>German 2006</td>
<td>22.0</td>
<td>23.4</td>
<td>28.6</td>
<td>39.1</td>
<td>56.2</td>
<td>44.8</td>
<td>52.4</td>
<td>45.0</td>
<td></td>
<td>41.2</td>
<td>21.8</td>
<td></td>
</tr>
<tr>
<td>Greek 2007</td>
<td>19.7</td>
<td>31.4</td>
<td>21.2</td>
<td>26.9</td>
<td>45.4</td>
<td>20.2</td>
<td>26.3</td>
<td>34.5</td>
<td>38.9</td>
<td>39.7</td>
<td>33.4</td>
<td></td>
</tr>
<tr>
<td>Hungarian 2007</td>
<td>5.5</td>
<td>41.4</td>
<td>8.0</td>
<td>58.2</td>
<td>58.3</td>
<td>51.8</td>
<td>34.0</td>
<td>34.1</td>
<td></td>
<td>24.1</td>
<td>48.1</td>
<td></td>
</tr>
<tr>
<td>Italian 2007</td>
<td>37.4</td>
<td>21.6</td>
<td>28.8</td>
<td>40.7</td>
<td>34.9</td>
<td>52.3</td>
<td>39.4</td>
<td>51.0</td>
<td></td>
<td>41.4</td>
<td>60.6</td>
<td></td>
</tr>
<tr>
<td>Japanese 2006</td>
<td>13.8</td>
<td>67.2</td>
<td>27.5</td>
<td>22.7</td>
<td>63.0</td>
<td>50.8</td>
<td>61.2</td>
<td>61.9</td>
<td>61.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portuguese 2006</td>
<td>31.2</td>
<td>25.8</td>
<td>33.5</td>
<td>72.4</td>
<td>74.5</td>
<td>50.6</td>
<td>69.6</td>
<td>75.1</td>
<td>51.5</td>
<td>55.8</td>
<td>31.4</td>
<td>41.8</td>
</tr>
<tr>
<td>Slovenian 2006</td>
<td>26.6</td>
<td>24.3</td>
<td>31.2</td>
<td>35.2</td>
<td>50.9</td>
<td>18.1</td>
<td>35.7</td>
<td>47.3</td>
<td>36.0</td>
<td>22.4</td>
<td>18.3</td>
<td>34.6</td>
</tr>
<tr>
<td>Swedish 2006</td>
<td>27.8</td>
<td>25.9</td>
<td>46.4</td>
<td>50.7</td>
<td>49.7</td>
<td>48.2</td>
<td>54.5</td>
<td>55.6</td>
<td>39.9</td>
<td>34.3</td>
<td>54.9</td>
<td>26.9</td>
</tr>
<tr>
<td>Spanish 2006</td>
<td>29.8</td>
<td>24.7</td>
<td>32.3</td>
<td>28.2</td>
<td>61.4</td>
<td>51.9</td>
<td>61.1</td>
<td>61.7</td>
<td>40.5</td>
<td>31.2</td>
<td>60.1</td>
<td>54.6</td>
</tr>
<tr>
<td>Turkish 2007</td>
<td>1.6</td>
<td>65.5</td>
<td>40.9</td>
<td>34.4</td>
<td>37.9</td>
<td>15.7</td>
<td>56.9</td>
<td>57.0</td>
<td>38.6</td>
<td>27.4</td>
<td>55.0</td>
<td>32.1</td>
</tr>
</tbody>
</table>

average: 26.4          | 29.8           | 33.2           | 42.9          | 48.6          | 40.0         | 48.7        | 50.0        | 43.3        | 43.0        | 38.9         | 39.4        |

Table 2: Results comparison on CoNLL datasets [4, 22]. Averaged numbers are sometimes not comparable since the average can be done over different set of languages.
### Table 3: Results comparison in the PASCAL Challenge on Grammar Induction [10].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>9.3</td>
<td>64.8</td>
<td>61.1</td>
<td>47.2</td>
<td>64.8</td>
<td>47.2</td>
<td>54.6</td>
<td>66.7</td>
</tr>
<tr>
<td>Basque</td>
<td>34.3</td>
<td>24.4</td>
<td>56.3</td>
<td>50.3</td>
<td>30.3</td>
<td>33.3</td>
<td>22.3</td>
<td>58.6</td>
</tr>
<tr>
<td>Czech</td>
<td>28.9</td>
<td>34.3</td>
<td>50.0</td>
<td>48.5</td>
<td>57.5</td>
<td>45.5</td>
<td>51.2</td>
<td>59.0</td>
</tr>
<tr>
<td>Danish</td>
<td>18.7</td>
<td>49.2</td>
<td>46.2</td>
<td>49.3</td>
<td>51.3</td>
<td>56.9</td>
<td>60.5</td>
<td>60.8</td>
</tr>
<tr>
<td>Dutch</td>
<td>34.0</td>
<td>39.5</td>
<td>50.5</td>
<td>50.8</td>
<td>49.5</td>
<td>38.9</td>
<td>50.0</td>
<td>51.7</td>
</tr>
<tr>
<td>English (childes)</td>
<td>36.0</td>
<td>23.3</td>
<td>48.1</td>
<td>62.2</td>
<td>47.2</td>
<td>50.5</td>
<td>53.5</td>
<td>56.0</td>
</tr>
<tr>
<td>English (PTB)</td>
<td>40.4</td>
<td>19.9</td>
<td>72.1</td>
<td>73.7</td>
<td>67.4</td>
<td>44.8</td>
<td>61.0</td>
<td>74.7</td>
</tr>
<tr>
<td>Portuguese</td>
<td>28.1</td>
<td>37.7</td>
<td>54.3</td>
<td>76.3</td>
<td>59.9</td>
<td>47.7</td>
<td>71.1</td>
<td>55.7</td>
</tr>
<tr>
<td>Slovenian</td>
<td>35.9</td>
<td>14.7</td>
<td>65.8</td>
<td>53.9</td>
<td>51.4</td>
<td>39.7</td>
<td>50.3</td>
<td>67.7</td>
</tr>
<tr>
<td>Swedish</td>
<td>33.2</td>
<td>31.3</td>
<td>65.8</td>
<td>66.7</td>
<td>63.7</td>
<td>48.2</td>
<td>72.0</td>
<td>76.5</td>
</tr>
<tr>
<td>Average</td>
<td>29.9</td>
<td>33.9</td>
<td>57.0</td>
<td>57.9</td>
<td>54.3</td>
<td>45.3</td>
<td>54.7</td>
<td>62.7</td>
</tr>
</tbody>
</table>


bust Unsupervised and Semisupervised Methods in Natural Language Processing, pages 1–8, Hissar, Bulgaria, 2011.


