Unsupervised Learning of Morphology for English and Inuktitut

Howard Johnson
Institute for Information Technology,
National Research Council
Howard.Johnson@nrc.gc.ca

Joel Martin
Institute for Information Technology
National Research Council
Joel.Martin@nrc.gc.ca

Abstract

We describe a simple unsupervised technique for learning morphology by identifying hubs in an automaton. For our purposes, a hub is a node in a graph with in-degree greater than one and out-degree greater than one. We create a word-trie, transform it into a minimal DFA, then identify hubs. Those hubs mark the boundary between root and suffix, achieving similar performance to more complex mixtures of techniques.

1 Introduction

To recognize a morpheme boundary, for example between a root and a suffix, a learner must have seen at least two roots with that suffix and at least two suffixes with that root. For instance, 'helpful', 'helpless', 'harmful', and 'harmless' would be enough evidence to guess that those words could be divided as 'help/ful', 'help/less', 'harm/ful', and 'harm/less'. Without seeing varying roots and varying suffixes, there is no reason to prefer one division to another.

We can represent a language's morphology as a graph or automaton, with the links labeled by characters and the nodes organizing which characters can occur after specific prefixes. In such an automaton, the morpheme boundaries would be *hubs*, that is, nodes with in-degree greater than one and out-degree greater than one. Furthermore, this automaton could be simplified by path compression to remove all nodes with in-degree and out-degree of one. The remaining automaton could be further modified to produce a graph with one source, one sink, and all other nodes would be hubs.

A *hub-automaton,* as described above, matches the intuitive idea that a language's morphology allows one to assemble a word by chaining morphemes together. This representation highlights the morphemes while also representing morphotactic information. Phonological information can be represented in the same graph but may be more economically represented in a separate transducer that can be composed with the hub-automaton.

For identifying the boundary between roots and suffixes, the idea of hubs is essentially the same as Goldsmith's (2001) *signatures* or the variations between Gaussier's (1999) p-similarity words. A signature is a set of suffixes, any of which can be added to several roots to create a word. For example, in English any suffix in the set: NULL, 's', 'ed', 'ing', can be added to 'want' or 'wander' to form a word. Here, NULL means the empty suffix.

In a hub automaton, the idea is more general than in previous work and applies to more complex morphologies, such as those for agglutinative or polysynthetic languages. In particular, we are interested in unsupervised learning of Inuktitut morphology in which a single lexical unit can often include a verb, two pronouns, adverbs, and temporal information.

In this paper, we describe a very simple technique for identifying hubs as a first step in building a hub-automaton. We show that, for English, this technique does as well as more complex collections of techniques using signatures. We then show that the technique also works, in a limited way, for Inuktitut. We close with a discussion of the limitations and our plans for more complete learning of hub-automata.

2 Searching for hubs

The simplest way to build a graph from a raw corpus of words is to construct a *trie*. A trie is a tree representation of the distinct words with a character label on each branch. The trie can be transformed into a minimal, acyclic DFA (deterministic finite automaton), sharing nodes that have identical continuations. There are well known algorithms for doing this (Hopcroft & Ullman, 1969). For example, suppose that, in a given corpus, the prefix 'friend' occurs only with the suffixes 'NULL', 's', and 'ly' and the word 'kind' occurs only with the same suffixes. The minimal DFA has merged the nodes that represent those suffixes, and as a result has fewer links and fewer nodes than the original trie.

In this DFA, some hubs will be obvious, such as for the previous example. These are morpheme boundaries. There will be other nodes that are not obvious hubs. Some may have high out-degree but an in-degree of one; others will have high in-degree but an out-degree of one.

Many researchers, including Schone and Jurafsky (2000), Harris (1958), and Déjean (1998), suggest looking for nodes with high branching (out-degree) or a large number of continuations. That technique is also used as the first step in Goldsmith's (2001) search for signatures. However, without further processing, such nodes are not reliable morpheme boundaries.

Other candidate hubs are those nodes with high out-degree that are direct descendants, along a single path, of a node with high in-degree. In essence, these are stretched hubs. Figure 1 shows an idealized view of a hub and a stretched hub.

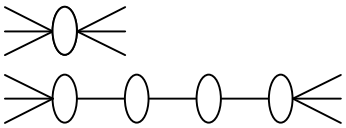

Figure 1: An idealized view of a hub and a stretched hub. The lines are links in the automaton and each would be labeled with a character. The ovals are nodes and are only branching points.

In a minimized DFA of the words in a corpus, we can identify hubs and the last node in stretched hubs as morpheme boundaries. These roughly correspond to the signatures found by other methods.

The above-mentioned technique for hub searching misses boundaries if a particular signature only appears once in a corpus. For instance, the signature for 'help' might be 'ed', 's', 'less', 'lessly', and NULL; and suppose there is no other word in the corpus with the same signature. The morpheme boundaries 'help-less' and 'help-ed' will not be found.

The way to generalize the hub-automaton to include words that were never seen is to merge hubs. This is a complex task in general. In this paper, we propose a very simple method. We suggest merging each node that is a final state (at the end of a word) with each hub or stretched hub that has in-degree greater than two. Doing so sharply increases the number of words accepted by the automaton. It will identify more correct morpheme boundaries at the expense of including some non-words.

These two techniques, hub searching and simple node merging, were implemented in a program called "*HubMorph*" (hub-automaton morphology).

3 Related Work

Most previous work in unsupervised learning of morphology has focused on learning the division between roots and suffixes (e.g., Sproat, 1992; Gaussier, 1999; Déjean, 1996; Goldsmith, 2001). The hope is that the same techniques will work for extracting prefixes. However, even that will not handle the complex combinations of infixes that are possible in agglutinative languages like Turkish or polysynthetic languages like Inuktitut.

This paper presents a generalization of one class of techniques that search for signatures or positions in a trie with a large branching factor. Goldsmith (2001) presents a well-developed and robust version of this class and has made his system, *Linguistica*, freely available (Goldsmith, 2002).

Linguistica applies a wide array of techniques including heuristics and the application of the principle of Minimum Description Length (MDL) to find the best division of words into roots and suffixes, as well as prefixes in some cases. The first of these techniques finds the points in a word with the highest number of possible successors in other words. With all these techniques, Linguistica seeks optimal breakpoints in each word. In this case, optimal means the minimal number of bits necessary to encode the whole collection.

There are also techniques that attempt to use semantic cues, arguing that knowing the signatures is not sufficient for the task. For example, Yarowsky and Wicentowski (2000; cf. Schone & Jurafsky, 2000) present a method for determining whether *singed* can be split into *sing* and *ed* based on whether *singed* and *sing* appear in the same contexts. Adopting a technique like this would increase the precision of HubMorph. In addition, some semantic approach is absolutely essential for identifying fusional morphology, where the word (*sang*) is not a simple composition of a root (*sing*) and morphemes.

4 Evaluation

As noted above, Linguistica uses many techniques to learn morphology, including a fairly complex system for counting bits. We tested whether the two techniques presented in this paper, hub searching and simple node merging, achieve the same performance as Linguistica. If so, the simpler techniques might be preferred. Also, we would be justified using them for more complex morphologies.

The input to *Linguistica* and *HubMorph* was the text of *Tom Sawyer*. The performance of both was compared against a gold standard division of the distinct words in that novel. The gold standard was based on dictionary entries and the judgment of two English speakers.

In matching the gold standard words to divisions predicted by either system, we made the following assumptions. a) Words with hyphens are split at the hyphen to match *Linguistica*'s assumption. b) If the gold standard has a break before and after a single character, to capture non-concatenative modification, either break matches. An example would be 'mud-d-y'. c) An apostrophe at a morpheme boundary is ignored for compari-

son matching to allow it to stick to the root or to the suffix. d) The suffix split proposed must result in a suffix of 5 or fewer characters, again to match *Linguistica*'s assumption.

Table 1 show the results of this comparison for *Linguistica*, hub-searching alone, and *HubMorph* (both hub searching and node merging). Hub-searching alone is sufficient to achieve the same precision as *Linguistica* and nearly the same recall. Both of the techniques together are sufficient to achieve the same precision and recall as *Linguistica*. The recall for all is low because the list of words in Tom Sawyer is not long enough to include most acceptable combinations of roots and suffixes. A longer input word list would improve this score.

System	Recall	Precision
Linguistica	0.5753	0.9059
Hub-Searching	0.4451	0.9189
HubMorph	0.5904	0.9215

Table 1: The recall and precision of *Linguistica*, Hub-searching alone, and *HubMorph*. Recall is the proportion of distinct words from Tom Sawyer that are correctly divided into root and suffix. Precision is the proportion of predicted divisions that are correct.

5 Discussion

HubMorph achieves the same performance as Linguistica on the words in Tom Sawyer. It does so with a general technique based on building a hub-automaton. In addition to being simple, HubMorph can be generalized to deal with more complex morphologies.

We have applied *HubMorph* to Inuktitut for dividing such words as ikajuqtaulauqsimajunga ("I was helped in the recent past", ikajuq-tau-lauq-sima-junga). The path in a hub automaton for most Inuktitut words would have many hubs, because the words have many divisions.

Currently, there are many limitations. The search for hubs in the middle of words is very difficult and requires merging nodes to induce new words. This will be necessary because Inuktitut theoretically has billions of words and only a small fraction of them has occurred in our source (the Nunavut, Canada Hansards).

Also, because each word has many morphemes, it is difficult to correctly detect the divisions for roots and suffixes. In general, there are no prefixes in Inuktitut, only infixes and suffixes.

Finally, there are many dialects of Inuktitut and many spelling variations. In general, the written language is phonetic and the spelling reflects all the variations in speech.

When HubMorph performs unsupervised learning of Inuktitut roots, it achieves a precision of 31.8% and a recall of 8.1%. It will be necessary to learn more of the infixes and suffixes to improve these scores.

We believe that hub-automata will be the basis of a general solution for IndoEuropean languages as well as for Inuktitut.

References

Déjean, H. 1998. *Morphemes as necessary concepts for structures: Discovery from untagged corpora.* University of Caen-Basse Normandie. http://citeseer.nj.nec.com/19299.html

Gaussier E. (1999). Unsupervised learning of derivational morphology from inflectional lexicons. In: Kehler A and Stolcke A, eds, *ACL workshop on Unsupervised Methods in Natural Language Learning*, College Park, MD.

Goldsmith, J.A. (2001). Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, 27:2 pp. 153-198.

Goldsmith, J.A. (2002). Linguistica software. http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000/.

Harris, Z. (1951). *Structural Linguistics*. University of Chicago Press.

Hopcroft, J.E. & Ullman, J.D. (1969). *Formal Languages and their Relation to Automata.* Addison-Wesley, Reading, MA.

Schone, P., & Jurafsky, D. (2000). Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of CoNLL-2000 and LLL-2000*, pp. 67--72 Lisbon, Portugal.

Sproat, R. (1992). *Morphology and Computation*, Cambridge, MA, MIT Press.

Yarowsky, D. & Wicentowski, R. (2000). Minimally supervised morphological analysis by multimodal alignment. In K. Vijay-Shanker and Chang-Ning Huang, editors, *Proceedings of the 38th Meeting of the Association for Computational Linguistics*, pages 207-216, Hong Kong.