Hapax: Probabilistic part-of-speech tagging in XQuery and XForms

C. M. Sperberg-McQueen
cmsmcq@blackmesatech.com
Black Mesa Technologies LLC, United States of America

Many programs perform part-of-speech (POS) tagging on texts [Leech et al. 1983, Booth 1985, Church 1988, DeRose 1988, Brill 1992, Leech et al. 1994, Schmidt 1994, 1995, Toutanova et al. 2003]; although they use a variety of algorithms, their interfaces tend to be similar:

- They work in batch mode, not interactively.
- They generally model text as a flat sequence of characters; for most, XML markup must be removed before data are submitted to the tagger, and afterwards merged back into the output.
- They are consequently unable to exploit information in XML markup — for example, that "Brown" is here a proper noun and "Essex" there a place name.
- They tag each word token in the input with their best guess at the correct POS; by default, they do not distinguish low- and high-probability guesses.
- They cannot accept partially tagged input. In consequence, the human annotator cannot help them by providing hints on some words.
- They operate on words, not smaller segments.

This paper describes an XQuery-based POS tagger designed to differ from existing taggers in all of these ways. It works interactively one sentence at a time directly on XML (by default, TEI-encoded) text, exploits relevant markup, provides not just the most probable tagging of the input but several ranked alternatives, accepts partially tagged input, and can work on user-specified segments (e.g. TEI w [word] or m [morph] elements) instead of only on space-delimited tokens. Because the tagger described here is designed to support semi-automatic (or 'half-automatic') POS annotation for XML data, it has been given the name Hapax.

Hapax has been designed and implemented as part of the project "Annotated Turki Manuscripts from the Jarring Collection Online" (ATMO), supported by the Henry R. Luce Foundation; the author thanks both the Luce Foundation for their support and his colleagues in the ATMO project for their collaboration.

Design considerations

Early POS taggers used morphological and other rules to assign POS tags to input; later experience showed that purely statistical methods like hidden Markov models (HMMs) could achieve better accuracy with less effort; for tutorial descriptions of HMMs see Rabiner 1989 and Charniak 1993.

For batch-mode POS tagging, accuracy and speed are obvious desiderata. Many modifications, refinements, and alternatives to HMMs have been proposed; these can improve accuracy by several percentage points. Larger training sets make a much larger difference. Schmid 1994 reports a comparison in which the least and most accurate taggers differ by two to four percentage points, while accuracy rates for small and large training sets (< 10,000 and > 1,000,000 words) differ by twelve to sixteen points.

For Hapax, intended to support human annotators working on under-resourced languages, raw speed is unimportant. For any tagger, the human annotator will need to correct many proposed taggings; the key to improving annotation speed is to make corrections faster.

Selecting the correct tag from a menu requires several interactions: the 80 tags in the Brown Corpus POS tag set do not fit into a single menu; many tag sets are larger. Accepting a proposed tagging for a word requires a single user-interface interaction (e.g. clicking "OK").

So speed improves with accuracy: the fastest corrections are those not needed. But high accuracy requires large training sets, which under-resourced languages lack by definition. Some algorithms cope well with limited data. In the Brown Corpus, 92% of all tokens are tagged with the most frequent POS tag for their word type. A trivial 1-gram tagger, which just assigns the most frequent POS tag for each word form, will thus do almost as well on known words as more sophisticated algorithms. In reality, not all words are known, but a 1-gram tagger trained on as little as 2000 words from the Brown Corpus will tag 60 to 70% of input tokens correctly. Larger training sets (8000, 32000, 128000, 500000 words) again do better (68-78%, 73-85%, 77-90%, 82-92%).

Also, we can make tagging errors less costly to fix. If the tagger provides one tag for each segment, every wrong guess costs a manual tag selection. If the tagger proposes several POS tags, then some errors will be as cheap as a correct tagging: one mouse-click. So the goal of Hapax's design is to minimize the need to select tags from menus, by proposing not one but several POS tags for each word.

If a 1-gram tagger for the Brown Corpus proposes not one but three POS tags, the correct tag will be among those proposed 71-80%, 79-86%, 84-93%, 87-97%, or 90-98% of the time (for 2000-, 8000-, 32000-, 128000, 500000-word training sets). If five tags are proposed, the correct tag will be proposed 79-88%, 87-92%, 91-95%, 92-97%, or 94-98% of the time.

If a single user interaction can accept a proposed tagging for the entire sentence, we will save one interaction for each word of the sentence. Hapax uses a standard bigram HMM to calculate the N most likely taggings for the entire sentence. The higher N is set, the greater the chances that only a single mouseclick will be required, but more time will be needed for reading and considering the proposals; it is likely that there is a point of diminishing returns.

XQuery implementation

Hapax is implemented as a library of XQuery functions. One set of functions reads the training material and produces XML word- or POS-frequency lists from them. These list word types or POS tags by frequency, subdivided by POS tags or word types (or, for bigrams, POS of following segment). Additional functions calculate probability distributions for use with unknown words, using the technique of Charniak et al. 1993.

The 1-gram tagger consults the word/POS frequency list and returns the N most likely POS tags for the given word form. The bigram tagger consults the bigram and POS/word lists and uses the standard Viterbi algorithm to calculate the most likely path through the trellis of possible taggings for a sentence. A simple modification of the algorithm allows Hapax to calculate not one path but the best N paths, with time linear in the number of tags in the trellis.

Testing routines generate random test and training sets from a corpus stored as an XQuery database; in a project setting, the training sets are not created on the fly but prepared in advance and stored in a database.The primary interface for consumers of the Hapax library is the function hapax:tag(), which accepts as arguments:

- An XML element representing a sentence
- An indication of what frequency data to use
- Optionally, a set of access functions

The function calls the 1-gram and bigram taggers and returns an XML document describing possible POS taggings for the input. In the common case, the input sentence is a tei:s element, containing tei:w or tei:m elements to be tagged. Input elements may have type attributes; such a partial tagging of the sentence will affect the probabilities for the POS tags for other elements. The optional set of access functions allows Hapax to be used with non-TEI markup; the user-supplied functions are used to identify words in a sentence, detect POS tagging in the input, and add POS tags to the output.

The entire Hapax library is a few thousand lines of XQuery; the rich sets of data structures (including XML as a native type), higher-order functions, and grouping constructs in XQuery and XSLT make the implementation of POS-tagging algorithms remarkably straightforward.

XForms interface

In the ATMO project, Hapax supports a browser-based user interface specified with XForms. The form displays a document, providing an Annotate button for each sentence. When the button fires, the form sends the sentence to the Hapax back end and uses the response to build a form for accepting or changing the annotation. The most likely taggings for the sentence are shown, each with an Accept button. A "Tag word-by-word" button is also shown; in word-by-word annotation, each segment in the sentence is displayed with several proposed tags: first those in the full-sentence taggings, then other common tags for the word type, and a worst-case "Tag manually" button which exposes the POS menus. The user can tag one or more words and activate a "Re-annotate" button, which resubmits the sentence to the back end. This allows the user to explore the effect of one POS assignment on POS probabilities for nearby words.

Within the ATMO project, data must also be segmented and spelling-regularized; those topics and their interaction with POS tagging are not discussed here.

Further work

Hapax v1 uses standard 1- and 2-gram HMMs for POS tagging (Charniak et al. 1993). Future versions should implement Schmid's binary-decision-tree method (1994, 1995), which helps with sparse data. More challenging will be adapting the directed-graph model of Xuehelaiti et al. (2013) to probabilistic POS

tagging. This two-level model would allow the probability of a given stem's POS tag to depend not only on the POS of the immediately preceding morpheme(s) but on the tag(s) of the preceding word stems, which may improve tagging accuracy for agglutinative languages.

Bibliography

Booth, B.M. (1985), "Revising CLAWS," *ICAME News* 9: 29-35.

Brill, E. (1992), "A simple rule-based part of speech tagger," in *Proceedings of the Third conference on applied natural language processing*, Trento 31 March - 3 April 1992 ([n.p.]: Association for Computational Linguistics), pp. 152-155.

Charniak, E. (1993), *Statistical language learning* (Cambridge: MIT Press).

Charniak, E., Hendrickson, C., Jacobson, N., and Parkowitz, M. (1993), "Equations for Part-of-Speech Tagging," in *Proceedings of the 11th National conference on artificial intelligence*, Washington DC July 11-15, 1993 ([n.p.]: The AAAI Press; Cambridge: MIT Press, 1993), pp. 784-789. Web. http://www.aaai.org/Papers/AAAI/1993AAAI93-117.pdf(Accessed: 1 October 2016)

Church, K. W. (1988), "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," in *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin 9-12 February 1988 ([n.p.]: Association for Computational Linguistics), pp. 136-143. Web. http://www.aclweb.org/anthology/A/A88/A88-1019.pdf. (Accessed: 1 October 2016)

DeRose, S.J. (1988), "Grammatical category disambiguation by statistical optimization," *Computational Linguistics* 14.1, pp. 31- 39.

Leech, G. Garside, R., and Bryant, M. (1994), "CLAWS4: The tagging of the British National Corpus," In *Proceedings of the 15th International conference on computational linguistics (COLING 94) Kyoto, Japan*, pp. 622-628. Web. http://ucrel.lancs.ac.uk/papers/coling.html . (Accessed: 1 October 2016)

Leech, G., Garside, R., and Atwell, E. (1983), "The automatic grammatical tagging of the LOB corpus," *ICAME News* 7: 13-33.

Rabiner, L. R. (1989) "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition."Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE* 77.2: 257-286.

Schmid, H. (1995), "Improvements in part-of-speech tagging with an application to German," In *Proceedings of the ACL SIGDAT-Workshop. Dublin, Ireland*. Revised version available on the Web at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf (Accessed: 1 October 2016.)

Schmid, H. (1994), "Probabilistic part-of-speech tagging using decision trees," In *Proceedings of International Conference on New Methods in Language Processing, Manchester, UK*. Web. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf . (Accessed: 1 October 2016)

Toutanova, K., Klein, D., Christopher Manning,, C. and Singer, Y. (2003), "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network," In *Proceedings of HLT-NAACL 2003,* pp. 252-259.

Xuehelaiti, M., Liu, K., Jiang, W., and Yibulayin, T.. (2013). "Graphic Language Model for Agglutinative Languages: Uyghur as Study Case." *Chinese computational linguistics and natural language processing based on naturally annotated big data: 12th China National Conference, CCL 2013 and First International Symposium, NLP-NABD 2013*, Suzhou, China, October 10-12, 2013, Proceedings, ed. Maosong Sun. LNAI 8202. Berlin: Springer, pp. 268-279.