Evaluation of Anchor Texts for Automated Link Discovery in Semi-structured Web Documents

Na'im Tyson, Jonathan Roberts, Jeff Allen, Matt Lipson
About.com
Sciences, 1500 Broadway, New York, NY, USA
ntyson@about.com, jroberts@about.com, jallen@about.com, mlipson@about.com

Abstract

Using an English noun phrase grammar defined by Hulth (2004a) as a starting point, we created an English noun phrase chunker to extract anchor text candidates identified within web-based articles. These phrases served as candidates for anchor texts linking articles within the About.com network of content sites. Freelance writers—serving as annotators with little to no training outside the domain authority of their respective fields—evaluated articles that received these machine-generated anchor texts using an annotation environment. Unlike other large-scale linguistic annotation projects, where annotators receive an evaluation based on a reference corpus, there was not sufficient time or funding to create a corpus of documents for anchor text comparisons amongst the annotators—thereby complicating the computation of inter-labeler agreement. Instead of using a reference corpus, we assumed that the anchor text generator was another annotator. We then computed the average Cohen's Kappa Coefficient (Landis and Koch, 1977) across all pairings of the anchor text generator and an annotator. Our approach showed a fair agreement level on average (as described in Pustejovsky and Stubbs (2013, p. 131–132)).

Keywords: kappa coefficient, chunking, chunk extraction, link discovery, anchor text

1. Introduction

About.com, also known as *The About Group*, publishes content for various subject domains from topicalized sites across seven major verticals: food, health, home, money, style, tech and travel. The website consists of almost 2 million articles that receive a monthly average of over 200 million visits from visitors primarily in the United States, Western Europe and parts of India. Experts write content in their domain of expertise; with the aid of a content management system, they select snippets of text as anchors to link to other relevant content in their own content website or throughout the entire About.com network.

Given that About.com is a publishing company that makes most of its revenue via advertising, we wish to keep users engaged by pointing them to different parts of the network for as long as possible. Inline links are a critical component of user recirculation—with higher clicks per session—compared to other recirculation methods on the site such as related article listings (at the bottom of an article), trending articles and navigation units around the website.

In our experience though, we found that our experts do not add as many inline links as they could during the process of creating their content. Producing quality links takes a great deal of time, and requires intimate knowledge of the full corpus of About.com content. Usually, experts are not cognizant of related articles written by experts outside of their own topical site. The histogram in Figure 1 demonstrates that the link density of articles (the number of About.com links in a given word count) is typically between 0 and 0.01 prior to the launch of automated link discovery on the site.

The solution was to build a tool that allows experts to select suggested anchor texts in their own articles and choose from the most suitable candidate destinations.

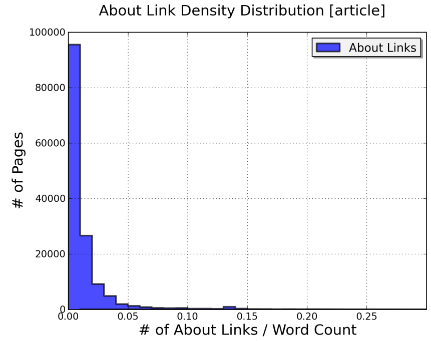

Figure 1: Histogram of link density of articles prior to the launch of automated link discovery.

2. Anchor Text Identification Process

2.1. Previous Approaches

Output from current keyword extraction techniques could serve as a basis for constructing anchor texts within an article given that both anchor texts and keywords encompass small spans of texts. Enhancing keyword extraction with part-of-speech information led to better quality keywords for a database of scientific journal papers (Hulth, 2004b). Other alternatives to linguistically-oriented keyword extraction systems such as KEA (Witten et al., 1999) and TextRank (Mihalcea and Tarau, 2004) might also work as well. The problem with all of these systems is that they tend to generate keyphrases between one and three words in length. In practice, the part-of-speech structures generated in expert-generated anchor texts—exemplified in Figure 2—can differ vastly from smaller noun phrase grammars proposed by Hulth (2004b) and other keyword extraction systems.

Another approach would be to use the existing link knowledge inside About.com to produce anchor and target can-

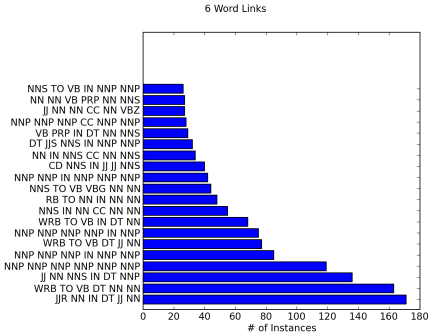

Figure 2: Part-of-speech (POS) histogram of expert-generated anchor texts consisting of six words in full-text articles.

didates. To calculate the strength of an anchor text, we could compute the *target strength* of an anchor text and target document as a ratio of the number of times an anchor points to the target document to the total number of times the anchor text appeared as a link (Erbs et al., 2011). However with so few documents on the site having a sufficient number of links, it would not be worthwhile to implement this technique.

Hence, we devised our own schema for selecting anchor text. All phrases needed to look natural without having any linguistically odd sequences. For example, in the phrase *The Rock and Roll Hall of Fame*, we would not want to use just *Rock and Roll* as a candidate anchor text. An additional requirement was that the method for generating anchor texts had to apply to all subject matter domains referenced on the website; it would be too cumbersome to create different grammar schemes of generating anchor texts for all of About.com's top-level verticals, and the sites existing within each one.

2.2. Methods for Generating Anchor Texts

2.2.1. Empirically-Driven Approach

Our original implementation had an empirical, linguistically-driven grammar to extract candidate anchor texts—where the grammar sequences originated from existing articles. Figure 2 illustrates an example distribution of grammar sequences. Although these grammar sequences produced longer sequences of anchor texts, they did not consistently identify named entities, and occasionally gave rise to nonsensical anchor texts. Some of these erratic anchor texts appeared due to errors in part of speech tagging.

2.2.2. Chunk Parsing for Anchor Text Generation

An intermediate solution would be to use compound grammatical structures that are less complex than the sequences illustrated earlier, yet general enough to identify potentially complicated grammatical structures. To this end, we used partial sentence parsing, otherwise known as *chunk parsing* (Abney, 1996), to extract phrases from a part-of-speech tagged sentence. Chunk extraction occurred via chunking rules, which are little more than regular expressions of tag sequences, implemented in Python's Natural Language Toolkit, NLTK (Loper and Bird, 2002). A noun phrase grammar defined by Hulth (2004a) served as a template for constructing the chunk rules, but it received a great deal of modification and expansion to handle the more complex tag sequences observed on About.com, which included named entities and date/time expressions.

The final anchor text candidates for a document were those having the maximum inverse document frequency across open class words comprising the entire phrase. Candidate destinations for the anchor texts were those having the highest document similarities between the anchor text, and a window of words around it. Because the primary focus of this evaluation was on the quality of the anchor texts, we will not concentrate on the exact method of computing similarity between source and target documents in this paper.

3. Evaluating Anchor Texts

3.1. Quality Assurance Setup

Before deploying automated link discovery throughout About.com, we decided to implement a Quality Assurance (QA) phase to adjust our algorithm for anchor text generation. This QA phase included 13 freelancers, who served as annotators, to verify anchor texts from approximately 86,000 articles chosen from our most highly viewed content on the site.

3.2. Annotation Workflow

In a similar fashion to Huang et al. (2009), where annotators had the opportunity to select link targets, and mark anchors and targets as relevant or irrelevant, our annotators had the following options within a web-based annotation environment: 1) keep an anchor text, 2) modify an anchor text by expanding or contracting it, 3) delete it entirely and 4) modify the link target. Annotators saw a single link target that they could delete, or supply one of their own. Usually annotators for tasks such as these would receive a great deal of training to ensure they could properly and consistently identify possible anchor texts in documents. In these circumstances, though, having few available options for the freelancers to mark up anchor texts and link targets inside the annotation environment made the need for further training somewhat of a burden—especially in light of the schedule to re-publish the documents with their enhanced links.

Freelancers received payment on an hourly basis, and did not garner additional wages upon the project's completion. The hourly incentive obviated the desire to annotate documents in haste. A database connected to the annotation environment tracked annotations across all of the freelancers' sessions; this gave content managers who managed the final documents the ability to undo certain annotations at some later time if they saw that the revisions were nonessential.

3.3. Evaluating Anchor Texts for Inter-Labeler Agreement

Evaluating the anchor texts in isolation proved to be a difficult task because the complete validation required some consideration of the link target. Assuming that the link target was satisfactory, then the previously mentioned options for altering the anchor text remain the same. If we used the entirety of the anchor text as the unit for evaluation, we fail to give credit to the generator when there is slight disagreement on the span of an anchor text. Consequently, the evaluation treats the anchor as a sequence of words to measure the relative agreement between the generator and an annotator. If the words in an anchor text remain unchanged, the relative agreement is one. We did not consider anchor texts with deleted link targets since we had no way of knowing why the annotator deleted the target link: the target link may be inappropriate for the anchor text, or the target link may not have fit the context of the article.

3.4. Computing Inter-labeler Agreement

Annotators were not privy to the anchor texts deemed unsuitable for linking by the generator, so there is no way to directly measure when both the generator and the annotator identified anchor texts as negative. As an approximation, each anchor text received a padding of one word before and after the text to estimate words that either the generator or annotator ignored. A caret and a dollar sign denoted the padded token at the beginning and end of phrases, respectively, as illustrated in Example 1.

Symbols a through d in the same example refer to cells in a contingency table, shown in Table 1, for each phrase extracted from a document. The letter 'A' denotes the generator, and 'B' represents an annotator; and the 'positive' label identifies an agreement between both annotators. Our assumption was that the annotator represented ground truth. The cell marked 'a' is the relative number of words where the generator and annotator agreed on the anchor text; we can consider this as the relative number of *true positive* words between the automatically generated anchor and the annotator's selection. Cell 'b' is a relative number of words that the generator suggested as anchor text, but the annotator modified or deleted it. These are the words that the generator falsely identified as anchor text and the annotator ignored—thereby making these words *false positives*.

When the generator did not select words in the anchor text, but the annotator inserted words, then we measured that relative disagreement in cell 'c', and called them *false negatives*. Padding tokens at the beginning and end of the anchor text selected by the algorithm and annotator—which indicate the outer boundaries of the anchor text—gave us the ability to approximate the number of *true negatives* between the algorithm and annotator for cell 'd'. Table 1 shows the placement of symbols a through d within a two-way contingency table; and Table 2 is an instantiation of Table 1 with the relative number of correct/incorrect words derived from all 11 words presented in Example 1.

Algorithm:		^	quick	brown	fox	jumps	over	the	lazy	dog	$
Annotator:	^	the	quick	brown	fox	$					
	d	c	a	a	a	b	b	b	b	b	d

Example 1: Example of phrase alighment between the anchor text generator and an annotator.

With a two-way contingency table for each pair of annotators, i.e., for the generator and the human annotator, we computed the *mean average precision*, MAP, and Cohen's Kappa (K) as shown in Equations 1 and 2, respectively.

		B	B
		positive	negative
A	positive	a	b
A	negative	c	d

Table 1: Contingency table for the anchor text generator (A), and a single annotator (B).

		B	B
		positive	negative
A	positive	3/11 = 0.27	5/11 = 0.45
A	negative	1/11 = 0.09	2/11 = 0.18

Table 2: Contingency table computed from relative word agreements from Table 1 for the generator (A) and annotator (B).

$$MAP = \frac{1}{|A|} \sum_{i=1}^{|A|} \frac{1}{m} \sum_{k=1}^{m} Precision(d_k) \tag{1}$$

$$K = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{2}$$

For the MAP calculation, we computed the average precision per document, $Precision(d_k)$, and then averaged across all annotators in the set A.[1] This gave us a MAP score of 0.40.

In Equation 2, $Pr(a)$ is the ratio of agreement between the annotators and the total number of annotations. In short, $Pr(e)$ is a summary metric of the expected agreement for each category label. It first involved calculating the percentage of times that the annotator used a particular label; this was the equivalent of summing across a row or column for a category and dividing by the total number of annotations. Since each annotator worked independently, we took the product across annotators. Finally, there was a summation of the product for each category across all categories because the distribution of the categories is disjoint.[2] Averaging over the kappa coefficients for the pairs of anchor text generator and annotator should yield a fair to moderate agreement level.

The average Kappa from all of the documents was 0.33, which would only be a *fair level of agreement* between the generator and the annotator. A preliminary chi-squared test on the relative agreements from the generator and the annotators showed that there was a difference between the relative agreements of the generator and annotators; therefore, the generator's choices are not always on par with those of the annotators.

[1] Precision is the number of true positions divided by the sum of true and false positives ($tp/tp + fp$).

[2] See Pustejovsky and Stubbs (2013, p. 133–134) for a detailed example of how to compute both $Pr(a)$ and $Pr(e)$.

4. Discussion

4.1. Incentivizing a Two-tier Approach to Web Page Annotation

In contrast to other Language Resource construction projects, where contributors/annotators have some social aspect to their work, the annotation tasks—as we described them here—are very solitary in nature. A freelancer logs into the annotation environment to validate anchor texts and linked targets, and nothing else. They received little to no feedback from the application or supervisor during the annotation process.

To ensure labeling consistency for the words comprising anchor texts, while promoting a social aspect to the process of annotation, we could divide the QA process into two subprocesses: a test for validation consistency and the actual QA process (as described in Section 3).

In a test of consistency, freelancers or content authors would receive the same set of documents for mark up. These documents would already have links inserted in them using the automated linking process; and this is no different than what we described earlier (but with far fewer documents). The difference here is that we would measure the *mean average relative agreement*, MAR, between each pair of annotators, and exclude the anchor text generator.

Using just the agreements a and d from Table 1 for an anchor text, t, we can compute the *average relative agreement* for a document consisting of each anchor text, t, with Equation 3:

$$\frac{1}{n}\sum_{i=1}^{n} a_{t_i} + d_{t_i} \tag{3}$$

Dividing the sum of a and d by the sum of a, b, c and d is not necessary since the agreements are relative, as shown in Table 2, and the denominator would have a sum close to one. Finally, we take the mean across the set of all documents, D, to compute the mean average relative agreement between any two annotators:

$$MAR = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{n}\sum_{i=1}^{n} a_{t_i} + d_{t_i} \tag{4}$$

With a MAR measurement for every pair of annotators, we can use a threshold to compare scores to find bad actors within the group (Neuendorf, 2002). If the MAR regularly falls below a threshold, the annotator would not receive an extra incentive to continue with the rest of the annotation of the corpus. Content managers could hold a general meeting among the annotators in order to expose good and bad practices in annotation, and allow the annotators to meet face-to-face.

4.2. Improvements to Anchor Text Selection

Thus far, we focused on honing the QA process to increase annotator consistency and compensation. To raise the level of agreement between the human annotators and the anchor text generator, there are a few options we can explore for enhancing the generator.

First, we could offer more parses of a sentence given the noun phrase grammar we constructed. NLTK returns a single parse of the sentence that matches the first rule within our noun phrase grammar. We could submit a pull request to the NLTK GitHub Project that fixes this issue. This requires a long-term commitment that we have to schedule into a future software release.

An alternative to this massive software enhancement would be to build a probabilistic noun phrase grammar in NLTK. Such an effort entails computing probabilities of noun phrase constructions from the existing anchor texts that already exist on the site. If there was not a sufficient number of examples for each noun phrase construction, we could turn to the anchor texts used as a result of the annotations, along with smoothed probabilities to accommodate those constructions where there were still not enough examples within the corpus.

5. Conclusions

We presented a novel framework for evaluating anchor texts generated by an automated link discovery system for the purpose of computing inter-labeler agreement. This evaluation scheme yielded only a fair level of agreement between the anchor text generator and the annotators we employed during the quality assurance phase of the automated link discovery system. With a reference corpus and better incentives offered to the annotators, accompanied by enhancements to the anchor text generation process, we hope to achieve a higher level of agreement in the foreseeable future.

6. Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments.

7. Bibliographical References

Abney, S. (1996). Tagging and partial parsing. In Ken Church, et al., editors, *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, Dordrecht.

Erbs, N., Zesch, T., and Gurevych, I. (2011). Link discovery: A comprehensive analysis. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 83–86, Sept.

Huang, W. C., Trotman, A., and Geva, S. (2009). The importance of manual assessment in link discovery. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 698–699, New York, NY, USA. ACM.

Hulth, A. (2004a). *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction*. Ph.D. thesis, Department of Computer and Systems Sciences, Stockholm University.

Hulth, A. (2004b). Enhancing linguistically oriented automatic keyword extraction. In *Proceedings of the Human Language Technology Conference*. North American

Chapter of the Association for Computational Linguistics.

Landis, J. and Koch, G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing Order into Texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

Neuendorf, K. A. (2002). *The content analysis guidebook*. Thousand Oaks, California: Sage Publications.

Pustejovsky, J. and Stubbs, A. (2013). *Natural Language Annotation for Machine Learning*. O'Reilly Media, Inc.

Witten, I., Paynter, G., Frank, E., Gutwin, C., and Nevill-Manning, C. (1999). Kea: Practical automatic keyphrase extraction. In *International Workshop on Description Logics*, pages 254–256.