For the sake of simplicity:
Unsupervised extraction of lexical simplifications from Wikipedia

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil and Lillian Lee

my89@cornell.edu, bopang@yahoo-inc.com, cristian@cs.cornell.edu, llee@cs.cornell.edu

Abstract

We report on work in progress on extracting lexical simplifications (e.g., "*collaborate*" → "*work together*"), focusing on utilizing edit histories in Simple English Wikipedia for this task. We consider two main approaches: (1) deriving simplification probabilities via an edit model that accounts for a mixture of different operations, and (2) using metadata to focus on edits that are more likely to be simplification operations. We find our methods to outperform a reasonable baseline and yield many high-quality lexical simplifications not included in an independently-created manually prepared list.

1 Introduction

Nothing is more simple than greatness; indeed, to be simple is to be great. —Emerson, *Literary Ethics*

Style is an important aspect of information presentation; indeed, different contexts call for different styles. Here, we consider an important dimension of style, namely, *simplicity*. Systems that can rewrite text into simpler versions promise to make information available to a broader audience, such as non-native speakers, children, laypeople, and so on.

One major effort to produce such text is the Simple English Wikipedia (henceforth SimpleEW)[1], a sort of spin-off of the well-known English Wikipedia (henceforth ComplexEW) where human editors enforce simplicity of language through rewriting. The crux of our proposal is to learn lexical simplifications from SimpleEW edit histories, thus leveraging the efforts of the 18K pseudonymous individuals who work on SimpleEW. Importantly, not all the changes on SimpleEW are simplifications; we thus also make use of ComplexEW edits to filter out non-simplifications.

Related work and related problems Previous work usually involves general syntactic-level transformation rules [1, 9, 10].[2] In contrast, we explore data-driven methods to learn *lexical simplifications* (e.g., "*collaborate*" → "*work together*"), which are highly specific to the lexical items involved and thus cannot be captured by a few general rules.

Simplification is strongly related to but distinct from paraphrasing and machine translation (MT). While it can be considered a directional form of the former, it differs in spirit because simplification must trade off meaning preservation (central to paraphrasing) against complexity reduction (not a consideration in paraphrasing). Simplification can also be considered to be a form of MT in which the two "languages" in question are highly related. However, note that ComplexEW and SimpleEW do not together constitute a clean parallel corpus, but rather an extremely noisy comparable corpus. For example, Complex/Simple same-topic document pairs are often written completely independently of each other, and even when it is possible to get good sentence alignments between them, the sentence pairs may reflect operations other than simplification, such as corrections, additions, or edit spam.

Our work joins others in using Wikipedia revisions to learn interesting types of directional lexical relations, e.g, "eggcorns"[3] [7] and entailments [8].

2 Method

As mentioned above, a key idea in our work is to utilize SimpleEW edits. The primary difficulty in working with these modifications is that they include not only simplifications but also edits that serve other functions, such as spam removal or correction of grammar or factual content ("fixes"). We describe two main approaches to this problem: a probabilistic model that captures this mixture of different edit operations (§2.1), and the use of *metadata* to filter out undesirable revisions (§2.2).

---

[1] http://simple.wikipedia.org

[2] One exception [5] changes verb tense and replaces pronouns. Other lexical-level work focuses on medical text [4, 2], or uses frequency-filtered WordNet synonyms [3].

[3] A type of lexical corruption, e.g., "acorn"→"eggcorn".

2.1 Edit model

We say that the kth article in a Wikipedia corresponds to (among other things) a title or *topic* (e.g., "Cat") and a sequence $\vec{d}_k$ of article versions caused by successive edits. For a given lexical item or phrase A, we write $A \in \vec{d}_k$ if there is any version in $\vec{d}_k$ that contains A. From each $\vec{d}_k$ we extract a collection $e_k = (e_{k,1}, e_{k,2}, \ldots, e_{k,n_k})$ of *lexical edit instances*, *repeats allowed*, where $e_{k,i} = A \rightarrow a$ means that phrase A in one version was changed to a in the next, $A \neq a$; e.g., "*stands for*" $\rightarrow$ "*is the same as*". (We defer detailed description of how we extract lexical instances from data to §3.1.) We denote the collection of $\vec{d}_k$ in ComplexEW and SimpleEW as C and S, respectively.

There are at least four possible edit operations: *fix* (o_1), *simplify* (o_2), *no-op* (o_3), or *spam* (o_4). However, for this initial work we assume $P(o_4) = 0$.[4]

Let $P(o_i \mid A)$ be the probability that o_i is applied to A, and $P(a \mid A, o_i)$ be the probability of $A \rightarrow a$ given that the operation is o_i. The key quantities of interest are $P(o_2 \mid A)$ in S, which is the probability that A should be simplified, and $P(a \mid A, o_2)$, which yields proper simplifications of A. We start with an equation that models the probability that a phrase A is edited into a:

$$P(a \mid A) = \sum_{o_i \in \Omega} P(o_i \mid A) P(a \mid A, o_i), \quad (1)$$

where Ω is the set of edit operations. This involves the desired parameters, which we solve for by estimating the others from data, as described next.

Estimation Note that $P(a \mid A, o_3) = 0$ if $A \neq a$. Thus, if we have estimates for o_1-related probabilities, we can derive o_2-related probabilities via Equation 1. To begin with, we make the working assumption that occurrences of simplification in ComplexEW are negligible in comparison to fixes. Since we are also currently ignoring edit spam, we thus assume that only o_1 edits occur in ComplexEW.[5]

Let $f_C(A)$ be the fraction of $\vec{d}_k$ in C containing A in which A is modified:

$$f_C(A) = \frac{|\{\vec{d}_k \in C | \exists a,i \text{ such that } e_{k,i} = A \rightarrow a\}|}{|\{\vec{d}_k \in C | A \in \vec{d}_k\}|}.$$

We similarly define $f_S(A)$ on $\vec{d}_k$ in S. Note that we count topics (version sequences), not individual versions: if A appears at some point and is not edited until 50 revisions later, we should *not* conclude that A is unlikely to be rewritten; for example, the intervening revisions could all be minor additions, or part of an edit war.

If we assume that the probability of any particular fix operation being applied in SimpleEW is proportional to that in ComplexEW— e.g., the SimpleEW fix rate might be dampened because already-edited ComplexEW articles are copied over — we have[6]

$$\widehat{P}(o_1 \mid A) = \alpha f_C(A)$$

where $0 \leq \alpha \leq 1$. Note that in SimpleEW,

$$P(o_1 \vee o_2 \mid A) = P(o_1 \mid A) + P(o_2 \mid A),$$

where $P(o_1 \vee o_2 \mid A)$ is the probability that A is changed to a different word in SimpleEW, which we estimate as $\widehat{P}(o_1 \vee o_2 \mid A) = f_S(A)$. We then set

$$\boxed{\widehat{\mathbf{P}}(\mathbf{o_2} \mid \mathbf{A}) = \max\left(\mathbf{0}, \mathbf{f_S(A)} - \alpha \mathbf{f_C(A)}\right).}$$

Next, under our working assumption, we estimate the probability of A being changed to a as a fix by the proportion of ComplexEW edit instances that rewrite A to a:

$$\widehat{P}(a \mid A, o_1) = \frac{|\{(k,i) \text{ pairs} \mid e_{k,i} = A \rightarrow a \wedge \vec{d}_k \in C\}|}{\sum_{a'} |\{(k,i) \text{ pairs} \mid e_{k,i} = A \rightarrow a' \wedge \vec{d}_k \in C\}|}.$$

A natural estimate for the conditional probability of A being rewritten to a under any operation type is based on observations of $A \rightarrow a$ in SimpleEW, since that is the corpus wherein both operations are assumed to occur:

$$\widehat{P}(a \mid A) = \frac{|\{(k,i) \text{ pairs} \mid e_{k,i} = A \rightarrow a \wedge \vec{d}_k \in S\}|}{\sum_{a'} |\{(k,i) \text{ pairs} \mid e_{k,i} = A \rightarrow a' \wedge \vec{d}_k \in S\}|}.$$

Thus, from (1) we get that for $A \neq a$:

$$\boxed{\widehat{\mathbf{P}}(\mathbf{a} \mid \mathbf{A}, \mathbf{o_2}) = \frac{\widehat{\mathbf{P}}(\mathbf{a} \mid \mathbf{A}) - \widehat{\mathbf{P}}(\mathbf{o_1} \mid \mathbf{A})\widehat{\mathbf{P}}(\mathbf{a} \mid \mathbf{A}, \mathbf{o_1})}{\widehat{\mathbf{P}}(\mathbf{o_2} \mid \mathbf{A})}.}$$

2.2 Metadata-based methods

Wiki editors have the option of associating a comment with each revision, and such comments sometimes indicate the intent of the revision. We therefore sought to use comments to identify "trusted"

---

[4] Spam/vandalism detection is a direction for future work.

[5] This assumption also provides useful constraints to EM, which we plan to apply in the future, by reducing the number of parameter settings yielding the same likelihood.

[6] Throughout, "hats" denote estimates.

revisions wherein the extracted lexical edit instances (see §3.1) would be likely to be simplifications.

Let $\vec{r}_k = (r_k^1, \ldots, r_k^i, \ldots)$ be the sequence of revisions for the kth article in SimpleEW, where r_k^i is the set of lexical edit instances ($A \rightarrow a$) extracted from the ith modification of the document. Let c_k^i be the comment that accompanies r_k^i, and conversely, let $R(Set) = \{r_k^i | c_k^i \in Set\}$.

We start with a seed set of trusted comments, $Seed$. To initialize it, we manually inspected a small sample of the 700K+ SimpleEW revisions that bear comments, and found that comments containing a word matching the regular expression *simpl* (e.g, "simplify") seem promising. We thus set $Seed :=$ $\{*\mathsf{simpl}*\}$ (abusing notation).

The SIMPL method Given a set of trusted revisions $TRev$ (in our case $TRev = R(Seed)$), we score each $A \rightarrow a \in TRev$ by the point-wise mutual information (PMI) between A and a.[7] We write RANK($TRev$) to denote the PMI-based ranking of $A \rightarrow a \in TRev$, and use SIMPL to denote our most basic ranking method, RANK($R(Seed)$).

Two ideas for bootstrapping We also considered bootstrapping as a way to be able to utilize revisions whose comments are not in the initial $Seed$ set.

Our first idea was to iteratively expand the set of trusted comments to include those that most often accompany already highly ranked simplifications. Unfortunately, our initial implementations involved many parameters (upper and lower comment-frequency thresholds, number of highly ranked simplifications to consider, number of comments to add per iteration), making it relatively difficult to tune; we thus omit its results.

Our second idea was to iteratively expand the set of trusted revisions, adding those that contain already highly ranked simplifications. While our initial implementation had fewer parameters than the method sketched above, it tended to terminate quickly, so that not many new simplifications were found; so, again, we do not report results here.

An important direction for future work is to differentially weight the edit instances within a revision, as opposed to placing equal trust in all of them; this could prevent our bootstrapping methods from giving common fixes (e.g., "*a*" $\rightarrow$ "*the*") high scores.

3 Evaluation[8]

3.1 Data

We obtained the revision histories of both SimpleEW (November 2009 snapshot) and ComplexEW (January 2008 snapshot). In total, $\sim$1.5M revisions for 81733 SimpleEW articles were processed (only 30% involved textual changes). For ComplexEW, we processed $\sim$16M revisions for 19407 articles.

Extracting lexical edit instances. For each article, we aligned sentences in each pair of adjacent versions using tf-idf scores in a way similar to Nelken and Shieber [6] (this produced satisfying results because revisions tended to represent small changes). From the aligned sentence pairs, we obtained the aforementioned lexical edit instances $A \rightarrow a$. Since the focus of our study was not word alignment, we used a simple method that identified the longest differing segments (based on word boundaries) between each sentence, except that to prevent the extraction of entire (highly non-matching) sentences, we filtered out $A \rightarrow a$ pairs if either A or a contained more than five words.

3.2 Comparison points

Baselines RANDOM returns lexical edit instances drawn uniformly at random from among those extracted from SimpleEW. FREQUENT returns the most frequent lexical edit instances extracted from SimpleEW.

Dictionary of simplifications The SimpleEW editor "Spencerk" (Spencer Kelly) has assembled a list of simple words and simplifications using a combination of dictionaries and manual effort[9]. He provides a list of 17,900 simple words — words that do not need further simplification — and a list of 2000 transformation pairs. We did not use Spencerk's set as the gold standard because many transformations we found to be reasonable were not on his list. Instead, we measured our agreement with the list of transformations he assembled (SPLIST).

---

[7]PMI seemed to outperform raw frequency and conditional probability.

3.3 Preliminary results

The top 100 pairs from each system (edit model[10] and SIMPL and the two baselines) plus 100 randomly selected pairs from SPLIST were mixed and all presented in random order to three native English speakers and three non-native English speakers (all non-authors). Each pair was presented in random orientation (i.e., either as $A \rightarrow a$ or as $a \rightarrow A$), and the labels included "simpler", "more complex", "equal", "unrelated", and "?" ("hard to judge"). The first two labels correspond to simplifications for the orientations $A \rightarrow a$ and $a \rightarrow A$, respectively. Collapsing the 5 labels into "simplification", "not a simplification", and "?" yields reasonable agreement among the 3 native speakers ($\kappa = 0.69$; 75.3% of the time all three agreed on the same label). While we postulated that non-native speakers[11] might be more sensitive to what was simpler, we note that they disagreed more than the native speakers ($\kappa = 0.49$) and reported having to consult a dictionary. The native-speaker majority label was used in our evaluations.

Here are the results; "-x-y" means that x and y are the number of instances discarded from the precision calculation for having no majority label or majority label "?", respectively:

Method	Prec@100	# of pairs
SPLIST	86% (-0-0)	2000
Edit model	77% (-0-1)	1079
SIMPL	66% (-0-0)	2970
FREQUENT	17% (-1-7)	-
RANDOM	17% (-1-4)	-

Both baselines yielded very low precisions — clearly not all (frequent) edits in SimpleEW were simplifications. Furthermore, the edit model yielded higher precision than SIMPL for the top 100 pairs. (Note that we only examined one simplification per A for those A where $\widehat{P}(o_2 \mid A)$ was well-defined; thus "# of pairs" does not directly reflect the full potential recall that either method can achieve.) Both, however, produced many high-quality pairs (62% and 71% of the *correct* pairs) *not* included in SPLIST. We also found the pairs produced by these two systems to be complementary to each other. We

---

[10] We only considered those A such that $freq(A \rightarrow *) > 1 \wedge freq(A) > 100$ on both SimpleEW and ComplexEW. The final top 100 $A \rightarrow a$ pairs were those with As with the highest $P(o_2 \mid A)$. We set $\alpha = 1$.

[11] Native languages: Russian; Russian; Russian and Kazakh.

believe that these two approaches provide a good starting point for further explorations.

Finally, some examples of simplifications found by our methods: "*stands for*" $\rightarrow$ "*is the same as*", "*indigenous*" $\rightarrow$ "*native*", "*permitted*" $\rightarrow$ "*allowed*", "*concealed*" $\rightarrow$ "*hidden*", "*collapsed*" $\rightarrow$ "*fell down*", "*annually*" $\rightarrow$ "*every year*".

3.4 Future work

Further evaluation could include comparison with machine-translation and paraphrasing algorithms. It would be interesting to use our proposed estimates as initialization for EM-style iterative re-estimation. Another idea would be to estimate *simplification priors* based on a model of inherent lexical complexity; some possible starting points are number of syllables (which is used in various readability formulae) or word length.

References

[1] R. Chandrasekar, B. Srinivas. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 1997.

[2] L. Deléger, P. Zweigenbaum. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. *Workshop on Building and Using Comparable Corpora*, 2009.

[3] S. Devlin, J. Tait. The use of a psycholinguistic database in the simplification of text for aphasic readers. In *Linguistic Databases*, 1998.

[4] N. Elhadad, K. Sutaria. Mining a lexicon of technical terms and lay equivalents. *Workshop on BioNLP*, 2007.

[5] B. Beigman Klebanov, K. Knight, D. Marcu. Text simplification for information-seeking applications. *OTM Conferences*, 2004.

[6] R. Nelken, S. M. Shieber. Towards robust context-sensitive sentence alignment for monolingual corpora. *EACL*, 2006.

[7] R. Nelken, E. Yamangil. Mining Wikipedia's article revision history for training computational linguistics algorithms. *WikiAI*, 2008.

[8] E. Shnarch, L. Barak, I. Dagan. Extracting lexical reference rules from Wikipedia. *ACL*, 2009.

[9] A. Siddharthan, A. Nenkova, K. McKeown. Syntactic simplification for improving content selection in multi-document summarization. *COLING*, 2004.

[10] D. Vickrey, D. Koller. Sentence simplification for semantic role labeling/ *ACL*, 2008.