EMNLP@CPH: Is frequency all there is to simplicity?

Anders Johannsen, Héctor Martínez, Sigrid Klerke[†], Anders Søgaard
Centre for Language Technology
University of Copenhagen
{ajohannsen|alonso|soegaard}@hum.ku.dk
sigridklerke@gmail.com[†]

Abstract

Our system breaks down the problem of ranking a list of lexical substitutions according to how simple they are in a given context into a series of pairwise comparisons between candidates. For this we learn a binary classifier. As only very little training data is provided, we describe a procedure for generating artificial unlabeled data from Wordnet and a corpus and approach the classification task as a semi-supervised machine learning problem. We use a co-training procedure that lets each classifier increase the other classifier's training set with selected instances from an unlabeled data set. Our features include n-gram probabilities of candidate and context in a web corpus, distributional differences of candidate in a corpus of "easy" sentences and a corpus of normal sentences, syntactic complexity of documents that are similar to the given context, candidate length, and letter-wise recognizability of candidate as measured by a trigram character language model.

1 Introduction

This paper describes a system for the SemEval 2012 English Lexical Simplification shared task. The task description uses a loose definition of simplicity, defining "simple words" as "words that can be understood by a wide variety of people, including for example people with low literacy levels or some cognitive disability, children, and non-native speakers of English" (Specia et al., 2012).

Feature	r
Ngram$_{sf}$	0.33
Ngram$_{sf+1}$	0.27
Ngram$_{sf-1}$	0.27
Len$_{sf}$	-0.26
Len$_{max}$	-0.26
RI$_{proto(l)}$	-0.18
Syn$_{cn}$	-0.17
Syn$_{w}$	-0.17
Syn$_{cp}$	-0.17

Feature	r
RI$_{proto(f)}$	-0.15
Char$_{max}$	-0.14
RI$_{orig(l)}$	-0.11
Len$_{tokens}$	-0.10
Char$_{min}$	0.10
SW$_{freq}$	0.08
SW$_{LLR}$	0.07
Char$_{avg}$	-0.04

Table 1: Pearson's r correlations. The table shows the three highest correlated features per group, all of which are significant at the $p < 0.01$ level

2 Features

We model simplicity with a range of features divided into six groups. Five of these groups make use of the distributional hypothesis and rely on external corpora. We measure a candidate's distribution in terms of its lexical associations (RI), participation in syntactic structures (Syn), or corpus presence in order to assess its simplicity (Ngram, SW, Char). A single group, Len, measures intrinsic aspects of the substitution candidate, such as its length.

The substitution candidate is either an adjective, an adverb, a noun, or a verb, and all candidates within a list share the same part of speech. Because word class might influence simplicity, we allow our model to fit parameters specific to the candidate's part of speech by making a copy of the features for each part of speech which is active only when the candidate is in the given part of speech.

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 408–412,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

Simple Wikipedia (SW) These two features contain relative frequency counts of the substitution form in Simple English Wikipedia (SW$_{freq}$), and the log likelihood ratio of finding the word in the simple corpus to finding it in regular Wikipedia (SW$_{LLR}$)[1].

Word length (Len) This set of three features describes the length of the substitution form in characters (Len$_{sf}$), the length of the longest token (Len$_{max}$), and the length of the substitution form in tokens (Len$_{tokens}$). Word length is an integral part of common measures of text complexity, e.g in the English Flesch–Kincaid (Kincaid et al., 1975) in the form of syllable count, and in the Scandinavian LIX (Bjornsson, 1983).

Character trigram model (Char) These three features approximate the reading difficulty of a word in terms of the probabilities of its forming character trigrams, with special characters to mark word beginning and end. A word with an unusual combination of characters takes longer to read and is perceived as less simple (Ehri, 2005).

We calculate the minimum, average, and maximum trigram probability (Char$_{min}$, Char$_{avg}$, and Char$_{max}$).[2]

Web corpus N-gram (Ngram) These 12 features were obtained from a pre-built web-scale language model[3]. Features of the form Ngram$_{sf\pm i}$, where $0 < i < 4$, express the probability of seeing the substitution form together with the following (or previous) unigram, bigram, or trigram. Ngram$_{sf}$ is the probability of substitution form itself, a feature which also is the backbone of our frequency baseline.

Random Indexing (RI) These four features are obtained from measures taken from a word-to-word distributional semantic model. Random Indexing (RI) was chosen for efficiency reasons (Sahlgren, 2005). We include features describing the semantic distances between the candidate and the original form (RI$_{orig}$), and between the candidate and a prototype vector (RI$_{proto}$). For the distance between candidate and original, we hypothesize that annotators would prefer a synonym closer to the original form. A prototype distributional vector of a set of words is built by summing the individual word vectors, thus obtaining a representation that approximates the behavior of that class overall (Turney and Pantel, 2010). Longer distances indicate that the currently examined substitution is far from the shared meaning of all the synonyms, making it a less likely candidate. The features are included for both lemma and surface forms of the words.

Syntactic complexity (Syn) These 23 features measure the syntactic complexity of documents where the substitution candidate occurs. We used measures from (Lu, 2010) in which they describe 14 automatic measures of syntactic complexity calculated from frequency counts of 9 types of syntactic structures. This group of syntax-metric scores builds on two ideas.

First, syntactic complexity and word difficulty go together. A sentence with a complicated syntax is more likely to be made up of difficult words, and conversely, the probability that a word in a sentence is simple goes up when we know that the syntax of the sentence is uncomplicated. To model this we search for instances of the substitution candidates in the UKWAC corpus[4] and measure the syntactic complexity of the documents where they occur.

Second, the perceived simplicity of a word may change depending on the context. Consider the adjective “frigid”, which may be judged to be simpler than “gelid” if referring to temperature, but perhaps less simple than “ice-cold” when characterizing someone’s personality. These differences in word sense are taken into account by measuring the similarity between corpus documents and substitution contexts and use these values to provide a weighted average of the syntactic complexity measures.

3 Unlabeled data

The unlabeled data set was generated by a three-step procedure involving synonyms extracted from Wordnet[5] and sentences from the UKWAC corpus.

[1] Wikipedia dump obtained March 27, 2012. Date on the Simple Wikipedia dump is March 22, 2012.

[2] Trigram probabilities derived from Google T1 unigram counts.

[3] The “jun09/body” trigram model from Microsoft Web N-gram Services.

[4] http://wacky.sslmit.unibo.it/

[5] http://wordnet.princeton.edu/

1) **Collection**: Find synsets for unambigious lemmas in Wordnet. The synsets must have more than three synonyms. Search for the lemmas in the corpus. Generate unlabeled instances by replacing the lemma with each of its synonyms. 2) **Sampling**: In the unlabeled corpus, reduce the number of ranking problems per lemma to a maximum of 10. Sample from this pool while maintaining a distribution of part of speech similar to that of the trial and test set. 3) **Filtering**: Remove instances for which there are missing values in our features.

The unlabeled part of our final data set contains $n = 1783$ problems.

4 Ranking

We are given a number of ranking problems ($n = 300$ in the trial set and $n = 1710$ for the test data). Each of these consists of a text extract with a position marked for substitution, and a set of candidate substitutions.

4.1 Linear order

Let $\mathcal{X}^{(i)}$ be the substitution set for the i-th problem. We can then formalize the ranking problem by assuming that we have access to a set of (weighted) preference judgments, $w(a \prec b)$ for all $a, b \in \mathcal{X}^{(i)}$ such that $w(a \prec b)$ is the value of ranking item a ahead of b. The values are the confidence-weighted pair-wise decisions from our binary classifier. Our goal is then to establish a total order on $\mathcal{X}^{(i)}$ that maximizes the value of the non-violated judgments. This is an instance of the Linear Ordering Problem (Martí and Reinelt, 2011), which is known to be NP-hard. However, with problems of our size (maximum ten items in each ranking), we escape these complexity issues by a very narrow margin—10! ≈ 3.6 million means that the number of possible orderings is small enough to make it feasible to find the optimal one by exhaustive enumeration of all possibilities.

4.2 Binary classication

In order to turn our ranking problem into binary classification, we generate a new data set by enumerating all point-wise comparisons within a problem and for each apply a transformation function $\Phi(\mathbf{a}, \mathbf{b}) = \mathbf{a} - \mathbf{b}$. Thus each data point in the new set is the difference between the feature values of two candidates. This enables us to learn a binary classifier for the relation “ranks ahead of”.

We use the trial set for labeled training data L and, in a transductive manner, treat the test set as unlabeled data U_{test}. Further, we supplement the pool of unlabeled data with artificially generated instances U_{gen}, such that $U = U_{test} \cup U_{gen}$.

Using a co-training setup (Blum and Mitchell, 1998), we divide our features in two independent sets and train a large margin classifier[6] on each split. The classifiers then provide labels for data in the unlabeled set, adding the k most confidently labeled instances to the training data for the other classifier, an iterative process which continues until there is no unlabeled data left. At the end of the training we have two classifiers. The classification result is a mixture-of-experts: the most confident prediction of the two classifiers. Furthermore, as an upper-bound of the co-training procedure, we define an oracle that returns the correct answer whenever it is given by at least one classifier.

4.3 Ties

In many cases we have items a and b that tie—in which case both $a \prec b$ and $b \prec a$ are violated. We deal with these instances by omitting them from the training set and setting $w(a \prec b) = 0$. For the final ranking, our system makes no attempt to produce ties.

5 Experiments

In our experiments we vary feature-split, size of unlabeled data, and number of iterations. The first feature split, SYN–SW, pooled all syntactic complexity features and Wikipedia-based features in one view, with the remaining feature groups in another view. Our second feature split, SYN–CHAR–LEN, combined the syntactic complexity features with the character trigram language model features and the basic word length features. Both splits produced a pair of classifiers with similar performance—each had an F-score of around .73 and an oracle score of .87 on the trial set on the binary decision problem, and both splits performed equally on the ranking task.

[6] Liblinear with L1 penalty and L2 loss. Parameter settings were default. http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

System	All	N	V	R	A
MicrosoftFreq	0.449	0.367	0.456	0.487	0.493
Syn–SW$_f$	0.377	0.283	0.269	0.271	0.421
Syn–SW$_l$	0.425	0.355	**0.497**	0.408	0.425
Syn–Char–Len$_f$	0.377	0.284	0.469	0.270	0.421
Syn–Char–Len$_l$	0.435	0.362	0.481	0.465	0.439

Table 2: Performance on part of speech. Unlabeled set was U_{test}. Subscripts tell whether the scores are from the **f**irst or **l**ast iteration

With a large unlabeled data set available, the classifiers can avoid picking and labeling data points with a low certainty, at least initially. The assumption is that this will give us a higher quality training set. However, as can be seen in Figure 1, none of our systems are benefitting from the additional data. In fact, the systems learn more when the pool of unlabeled data is restricted to the test set.

Our submitted systems, Ord1 and Ord2 scored 0.405 and 0.393 on the test set, and 0.494 and 0.500 on the trial set. Following submission we adjusted a parameter[7] and re-ran each split with both U and U_{test}.

We analyzed the performance by part of speech and compared them to the frequency baseline as shown in Table 2. For the frequency baseline, performance is better on adverbs and adjectives alone, and somewhat worse on nouns. Both our systems benefit from co-training on all word classes. Syn–Char–Len, our best performing system, notably has a score reduction (compared to the baseline) of only 5% on adverbs, eliminates the score reduction on nouns, and effectively beats the baseline score on verbs with a 6% increase.

6 Discussion

The frequency baseline has proven very strong, and, as witnessed by the correlations in Table 1, frequency is by far the most powerful signal for "simplicity". But is that all there is to simplicity? Perhaps it is. For a person with normal reading ability, a simple word may be just a word with which the person is well-acquainted—one that he has seen before enough times to have a good idea about what it means and in which contexts it is typically used.

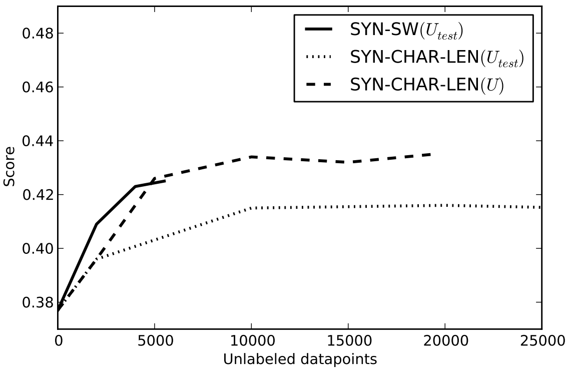

Figure 1: Test set kappa score vs. number of data points labeled during co-training

And so an n-gram model might be a fair approximation. However, lexical simplicity in English may still be something very different to readers with low literacy. For instance, the highly complex letter-to-sound mapping rules are likely to prevent such readers from arriving at the correct pronunciation of unseen words and thus frequent words with exceptional spelling patterns may not seem simple at all.

A source of misclassifications discovered in our error analysis is the fact that substituting candidates into the given contexts in a straight-forward manner can introduce syntactic errors. Fixing these can require significant revisions of the sentence, and yet the substitutions resulting in an ungrammatical sentence are sometimes still preferred to grammatical alternatives.[8] Here, scoring the substitution and the immediate context in a language model is of little use. Moreover, while these odd grammatical errors may be preferable to many non-native English speakers with adequate reading skills, such errors can be more obstructing to reading impaired users and beginning language learners.

Acknowledgments

This research is partially funded by the European Commission's 7th Framework Program under grant agreement n° 238405 (CLARA).

[7] In particular, we selected a larger value for the C parameter in the liblinear classifier.

[8] For example sentence 1528: "However, it appears they intend to *pull* out all stops to get what they want." Gold: {try everything} {do everything it takes} {pull} {stop at nothing} {go to any length} {yank}.

References

C. H. Bjornsson. 1983. Readability of Newspapers in 11 Languages. *Reading Research Quarterly*, 18(4):480–497.

A Blum and T Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Linnea C. Ehri. 2005. Learning to read words: Theory, findings, and issues. *Scientific Studies of Reading*, 9(2):167–188.

J P Kincaid, R P Fishburne, R L Rogers, and B S Chissom. 1975. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496.

Rafael Martí and Gerhard Reinelt. 2011. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization (Applied Mathematical Sciences)*. Springer.

Magnus Sahlgren. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5.

Lucia Specia, Sujay K. Jauhar, and Rada Mihalcea. 2012. SemEval-2012 Task 1: English Lexical Simplification. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, Montreal, Canada.

P. D Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.