Interactive Second Language Learning from News Websites

Abstract

We propose a web browser extension that allows readers to learn a second language vocabulary while reading news online. Injected tooltips allow readers to look up selected vocabulary and give interactive tests to assess vocabulary mastery.

We discover that two key system components needed improvement, both which stem from the need to model context. These two issues are in practical word sense disambigution (WSD) to aid translation quality and constructing the interactive tests. We start with Microsoft's Bing translation API but employ additional dictionary based heuristics that significantly improve translation quality over a baseline in both coverage and accuracy. We also propose techniques for generating appropriate distractors for multiple-choice word mastery tests. Our preliminary user survey confirms the need and viability of such a language learning platform.

1 Introduction

Learning a new language from language learning websites is time consuming. Research shows that regular practice, guessing, memorization (?),BUG as well as immersion into real scenarios (?)BUG hastens language learning process. To make second language learning attractive and efficient, we seek to interleave language learning with a popular daily activity: online news reading.

Most existing language learning software are either instruction-driven or user-driven. Duolingo[1] is a popular instruction-driven system that teaches through structured lessons. Instruction driven systems demand dedicated learner time on a daily basis and are limited by learning materials as lesson curation is often labor-intensive.

In contrast, many people informally use Google Translate[2] to learn vocabulary, making it a prominent example of a user-driven system. Translate, however, lacks the rigor of a learning platform as it lacks tests to allow learners to demonstrate mastery. In our work, we merge learning and assessment within the single activity of news reading. Our system also adapts to the learner's skill during assessment.

We propose a system to enable online news readers to efficiently learn a new language. Our prototype targets Chinese language learning while reading English language news. Learners are provided translations of open-domain words for learning from an English news page. In the same environment – for words that the system deems mastered by the learner – learners are assessed by replacing the original English text in the article with their Chinese translations and asked to translate them back given a choice of possible translations. The system, deployed as a Chrome web browser extension, is triggered when readers visit a preconfigured list of news websites.

A key design property of our language learning extension is only active on certain news websites. This is important as news articles typically are classified with respect to a news category, such as *finance*, *world news*, and *sports*. If we know which category of news the learner is viewing, we can leverage this contextual knowledge to improve the learning experience.

In the development of the system, we discovered two key components that can be affected by

[1] https://www.duolingo.com/

[2] https://translate.google.com/

this context modeling. We report on these developments here. In specific, we propose algorithms: **(i)** for translating English words to Chinese from news articles, **(ii)** for generating distractor translations for learner assessment.

2 The `SystemA` Chrome Extension

We give a running scenario to illustrate the use of our language learning platform, `SystemA`. When a learner browses to an English webpage on a news website, our extension selectively replaces certain original English words words with their Chinese translation (Figure 1, middle). While the meaning of the Chinese word is often apparent in context, the learner can choose to learn more about the replaced word, by mousing over the translation to reveal a definition tooltip (Figure 1, left) to aid mastery of the Chinese word. Once the learner has encountered the replaced word a few times, `SystemA` will assess the learner's mastery by generating a multipl choice translation test on the target word (Figure 1, right). Our learning platform thus can be viewed as have three logical components: *translating*, *learning* and *testing*.

Translating. We pass the main content of the webpage from the extension client to our server for candidate selection and translation. As certain words are polysemous, the server must select the most appropriate translation among all possible meanings. Our initial selection method replaces any instance of words stored in our dictionary. For translation, we check the word's stored meanings against the machine translation of each sentence obtained from the Microsoft Bing Translation API (hereafter, "Bing"). Matches are deemed as correct translations and are pushed back to the Chrome client for rendering.

Learning. Hovering the mouse over the replacement Chinese word causes a tooltip to appear, which gives the translation, pronunciation, simplified and traditional written form, and a `More` link that loads additional contextual example sentences (that were previously translated by the backend) for the learner to study. The more link must be clicked for activation, as we find this two-click architecture helps to minimize latency and the loading of unnecessary data. The server keeps record of the learning tooltip activations, logging the enclosing webpage URL, the target word and the user identity.

Testing. After the learner encounters the same word a pre-defined number t =BUG times, `SystemA` generates a MCQ test to assess mastery. When the learner hovers over the replaced word, the test is shown for the learner to select the correct answer. When an option is clicked, the server logs the selection, and the correct answer is revealed by the client extension. Statistics on the user's test history are also updated.

2.1 News Categories

As our learning platform is active only on certain news websites, we model the news category of both individual words and webpages. Of particular import to `SystemA` is the association of words to a news category, which is used downstream in both word sense disambiguation (Section **??**) and the generation of distractors in the interactive tests (Section 4. Here, our goal is to automatically find highly relevant words to a particular news category – e.g., "what are typical *finance* words?".

We first obtain a large sample of categorized English news webpages, by creating custom crawlers for specific news websites. We use a seed list of words that are matched against a target webpage's URL. If any match, the webpage is deemed to be of that category. For example, a webpage that has the seed word "football" in its URL is deemed of category "Sports". After a survey of a number of news websites, we decided on seven categories: namely, "World", "Technology", "Sports", "Entertainment", "Finance", "Health" and "Travel".

We tokenize and part-of-speech tag the main body text of the categorized articles, discarding punctuation and stopwords. The remaining words are classified to a news category based on document frequency. A word w is classified to a category c if it appears a tunable threshold δ =BUG more often than its average category document frequency. Note that a word can be categorized to multiple categories under this scheme.

3 Practical Word Sense Disambiguation

As we all know, one word often have multiple translations in another language, and our extension is expected to show the most appropriate one

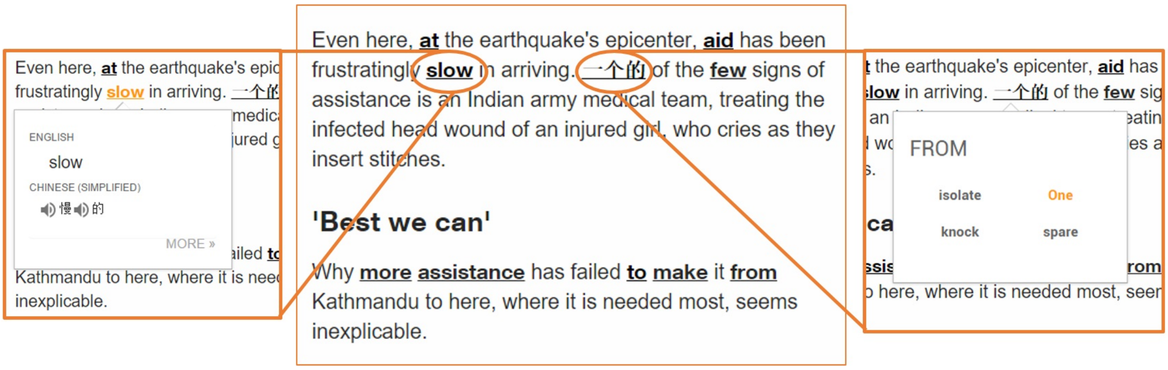

Figure 1: Merged screenshots of our Chrome extension on the CNN English article *Treacherous journey to epicenter of deadly Nepal earthquake*. Underlined components are clickable to yield tooltips of two different forms: (l) a definition for learning, (r) a multiple-choice interactive test.

based on the context. We call such translation selection as word sense disambiguation (WSD). WSD is an open task in natural language processing, aiming at identifying the proper sense (*i.e.*, meaning) of a word in a context, when the word has multiple meanings (Navigli, 2009). Traditionally, WSD system identifies the proper sense in the same language, while we show the proper sense in the form of another language.

In WSD, context information is the key to disambiguate word sense. We, therefore, make use of different granularity of context, *i.e.*, the category of the news, the word class, and the sentence, to select proper translations from our bilingual dictionary.

3.1 News Category

Topic information have been shown useful in WSD (Boyd-Graber et al., 2007). Take English word "interest" as an example. In finance related articles, "interest" is more likely to be "a share, right, or title in the ownership of property" ("利息" in Chinese), than 'the feeling of a person whose attention, concern, or curiosity is particularly engaged by something" ("兴趣"). Therefore, analysing the topic of the original article and selecting the translation with the same topic label might help disambiguate the word sense. We leverage the algorithm described in Section 2.1 to obtain the category for news and candidate Chinese translations.

3.2 Part-of-Speech Tagger

The word class, *i.e.*, the Part-of-Speech (POS) tag is believed to be beneficial for WSD (Wilks and Stevenson, 1998) and Machine Translation (Toutanova et al., 2002; Ueffing and Ney, 2003). For example, the English word "book" has two major classes, verb and noun, meaning "reserve" ("预定" in Chinese) and "printed work" ("书"), respectively. The two Chinese translations have the same POS tag as their corresponding English counterpart. Therefore, once knowing the POS tag for the English word in the context, we are able to pick up the Chinese translation with the same POS tag from the dictionary. In our system, we employ Standford Log-linear Part-of-Speech tagger (Toutanova et al., 2003) to obtain the POS tag for English word, and POS tag for Chinese words are contained in our dictionary. In some cases, after applying this rule, there is still multiple candidate Chinese translations and we will choose the most frequently used one.

3.3 Machine Translation

A richer context can be exploited is the neighboring word. We regard the whole sentence as the context for the target word and send the sentence to Microsoft Bing Translator[3], an online machine translation system with a limited free usage. The returned Chinese translation, however, does not have explicit word alignment to the original English sentence. Therefore, we need additional processing on the Chinese sentence, in order to find

[3] `https://www.bing.com/translator/`

Table 1: Example input and output of our word sense disambiguation configurations. **Boldface** indicates (Column 1) the target word to translate and, (Columns 3–8) the correct translation(s).

English fragment	Dict. entries	1. Baseline	2. Category	3. Bing	4. Bing+	5. Bing++
treating me **like** family	verb : 喜欢, 爱 preposition : 好像, 好比	喜欢	好像			
painting a **picture** of urban street life	相, 影, 影片(entertainment), 帧, 想象, 画		影片			
pistol a **pump** shotgun	verb:抽, 抽水, 打气, 唧, 唧筒, 套 noun:抽水机, 唧筒			唧筒		
have made it into the world's **top** 40 clubs	顶部, 顶端, 顶, 颠, 盖, 极	顶部		顶	顶级	
state department spokeswoman	陈, 陈说, 称, 称述, 发表, 发言			发言	发言人	国家

the Chinese word for the target English word.

Bing.

As all potential Chinese translations are available in our dictionary, the most intuitive processing is to perform a substring match, *i.e.*,

Bing+. Bing approach is not perfect. The results that generated by Bing approach is limited by the covearge of our dictionary size. In Table 1, the fourth example is the approach of using Bing Translator together with Stanford Word Segmenter, and I would like to use Bing+ to represent this algorithm. The Bing approach will generate "顶" as the result. After that, our algorithm will send the Chinese sentence returned from Bing Translator to Stanford Word Segmenter. Then, this algorithm will use the segmented word that contains the Bing result as a substring or equals to the Bing result as the final result. In this example, the final result of Bing+ is "顶级" which is the best result that can be generated from the result of Bing Translator and also a result that does not covered by our dictionary.

Bing++. Bing+ approach is not perfect as well. The results from Bing+ approach is highly related to the accuracy of string matching algorithm. If two English words shares very similar translations or if two Chinese words contains the same Chinese charater, Bing+ approach will generate the wrong result and that's why we need a Word Alignment tool.Bitext word alignment or simply word alignment is the natural language processing task of identifying translation relationships among the words (or more rarely multiword units) in a bitext, resulting in a bipartite graph between the two sides of the bitext, with an arc between two words if and only if they are translations of one another. I use Bing Word Alignment API[4] as our Word Alignment tool. The Bing++ algorithm is basically the approach of using Bing+ approach together with the Microsoft Bing Word Alignment. In Table 1, the fifth example, "state" is the word that need to be translated. The result from Bing+ approach is "发言人", which is the translation of "spokeswoman", because the Chinese translation "发言" can be translated from both "state" and "spokeswoman". Then step five will send the original English sentence to Bing Word Alignment. Now, there will be two final results, one from Bing+ approach and the other one from Bing Word Alignment and the algorithm will choose the correct one from these two results. In this example, "state" will match with "国家" and the algorithm will choose "国家" as the final result as well.

3.4 Evaluation

Our Word Sense Disambiguate System can be evaluated from two important aspects: coverage (i.e., is able to return a translation) and accuracy (i.e., the translation is proper). To this end, I manually annotate the ground truth.

Table 2 column two contains the coverage for different approaches. As the algorithm will try to translate some word only if it is covered by our dictionary, the coverage for Baseline is always 100%. The coverage for Bing, Bing+, Bing++ and

[4] `https://msdn.microsoft.com/en-us/library/dn198370.aspx`

Table 2: Experimental results.

	Coverage	Accuracy
Baseline	100%	57.3%
POSTagger	94.5%	55.2%
News Category	2.0%	7.1%
Bing	78.5%	79.8%
Bing+	75.7%	80.9%
Bing++	76.9%	97.4%

POSTagger are roughly the same and all of them are acceptable. However, the coverage for News Category approach is only 2.0%. One reason is that when I set the threshold for assigning categories for Chinese word, I purposely make it very high to maximize the accuracy. If the accuracy is quite high, which means this approach is quite useful, then I will lower the threshold and find the balance point.

Figure 2 column three contains the accuracy of all the approaches. The last column is the accuracy for News Category approach and it is only 7.1%. As mentioned in above Chapter, since the accuracy is very low, there is no need to lower the threshold and try to allocate more categories for Chinese words. The accuracy for Baseline is 57.3%, which is already a fairly hight accuracy. The accuracy for POSTagger is around 55.2% also, which is a bit lower than our expectation. The accuracy for Bing++ is 97.4% which I think is a very good result and it is already very hard to improve. Therefore, based on my test results, Bing++ is the best approach among these five approaches.

4 Distractors Generation Algorithm

The key research topic here is to investigate a way to automatically generate suitable distractors for a certain vocabulary test. The distractors are generated in English form.

4.1 Collecting category-related words

To generate good category-related distractors, it is essential to gather enough words that are more related in a certain category to serve as distractors candidates. By using the approach discussed in Section 3, we crawled more than 1400 articles for seven categories, with around 200 articles in each category. The confidence factor is selected to be 10, which is suitable to classify enough words into different categories. After this step, there should be sufficient "Category-Related" words in each category.

4.2 Generating distractors

The category-related words obtained from the previous step will be used in this step. Out selection strategy in choosing distractors takes following parameters:

- News website URL
- News sentence
- Word to test
- User's knowledge level of the word

Detect news category. After getting the news URL, our system needs to determine the category of the news. Based on the analysis from most popular news URLs, there is a set of common identifiers that can identify the category of the news article. For example, technology news URL often contains "/tech", "/science", and if we find these strings in news URL, we will classify this news URL into "Technology" category. The algorithm will go through all category identifier in the list, and will return the category name the moment it finds a match. The current list of category provides reasonable accuracy for the purpose of detecting news category.

Detect Part-Of-Speech Tag. Given the target word and the target sentence, it is easy to run the NLTK POS tagger to get the correct POS tag of this word. This step is essential to help select distractors with similar forms, i.e. if the target word is adjective, it will be appropriate to choose three other adjectives, not verbs, as distractors.

Semantic Distance. Before we go to explain the next step, it is essential to introduce the semantic distance calculator we used in the server implementation.

The perspective of semantic relatedness or its inverse, semantic distance, is a concept that indicates the likeness of two words. It is more general than the concept of similarity as stated in WordNet's synset relation. Similar entities in WordNet are classified into same synset based on their similarity. However, dissimilar entries may also have a close semantic connection by lexical relationships such as meronymy (car-wheel) and antonymy (hot-cold), or just by any kind of functional relationship or frequent association(pencil-paper, penguin-Antarctica) (Alexander, 2001). Semantic distance calculator aims to calculate the semantic relatedness score between two words.

There are many approaches to calculate semantic relatedness score. In this application, we are using Lin Distance (Lin, 1998) to calculate the semantic distance between two concepts. The detail of Lin Distance methodology is explained as follows.

Lin attempted to define a measure of semantic similarity that would be both universal and theoretically justified. There are three intuitions that he used as a basis:

- The similarity between arbitrary objects A and B is related to their commonality; the more commonality they share, the more similar they are;
- The similarity between A and B is related to the differences between them; the more differences they have, the less similar they are.
- The maximum similarity between A and B is reached when A and B are identical, no matter how much commonality they share.

Based on the intuition above, Lin proposed his approach in measuring similarity between two concepts c1, c2 in Equation 1:

$$sim(c1, c2) = \frac{2 * log_p(lso(c1, c2))}{log_p(c1) + log_p(c2)} \qquad (1)$$

where p(c) denotes the probability of encountering concept c, and lso(c1,c2) denotes the lowest common subsumer, which is the lowest node in WordNet hierarchy that is a hypernym of c1 and c2.

The distance calculator will return a score from 0 to 1, as can be easily seen from the formula above. If the score is closer to 1, it means the two words are closer in semantic sense. This distance calculator will play an important role in the following algorithm.

4.2.1 Distractors Selection Algorithm

Based on the input parameters, at this stage the server has already got the current category of the news article and the correct POS tag of the target word to test. The server is going to generate distractors based on user's knowledge level of the target word to test.

Knowledge level is 1: This indicates that the user has just learnt this word. The algorithm will randomly select three words from current category's word list. The reason for using randomization is to avoid the situation that similar distractors are generated every time.

Knowledge level is 2: This indicates that the user has known this word for some times. The algorithm will randomly select two words from the current category's word list as two distractors. Then the algorithm will randomly select word from the current category's word list and calculated the semantic distance between the selected word and the target word, once the score is above certain threshold, the selected word will be chose as the third distractor. The selection of threshold value will have a direct effect on the speed of distractors generation process. As a very high threshold value will result in more rounds of calculation in semantic distance calculator, and it will take a long time before the distractors are returned to the front end. After several rounds of analysis of each category's words and the results returned from semantic distance calculator, the threshold value of 0.1 is selected.

Knowledge level is 3: This indicates that the user has a good understanding of the word already; the algorithm will choose distractors solely based on results returned from semantic distance calculator. Similar to the approach when knowledge level is 2, the algorithm will randomly select word from current category's word list and calculate the semantic distance between the selected word and the target word. If the score is above certain threshold, the selected word is chosen as one of the distractors. The process is continued until the server can find three distractors.

4.3 Evaluation

To evaluate the distractors selection strategy as described in this report, we chose the knowledge-based approach used by many other language

learning systems, which is to utilize the WordNet data and selection distractors based on synonyms of synonyms. WordGap system uses this approach to generate vocabulary test for its android application.
In our implementation of the baseline algorithm, we will choose the most frequent used word w1 from the target word's synonym set, and select the most frequent used word w2 from word w1's synonym set. The selection process is continued until we can find 3 distractors to form a vocabulary test. However, if the number of valid result we can get is less than 3, we will choose the word that shares the same antonym with the target word.

4.3.1 Designing Survey

To compare the two approaches in generating distractors, we designed several survey sets to ask users to compare the plausibility of distractors. We randomly selected 50 sentences from recent news articles and choose one noun or adjective inside the sentence as the target word to test. In the survey, participants are required to answer each question and rank the plausibility of all distractors from 1 to 7. The correct answer will be ranked as 1, and the least plausible distractor will be ranked as 7. A screenshot of one sample question is shown in Figure 2.

There are two evaluations to be done as follows:

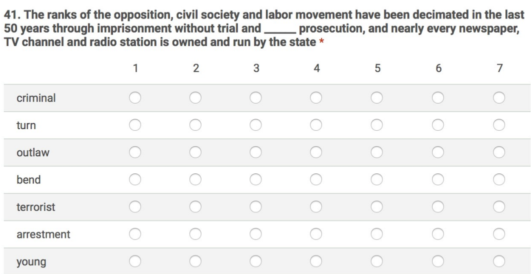

Figure 2: A sample survey question

1. Compare Baseline with Knowledge Level 1 Algorithm

. Compare Baseline with Knowledge Level 3 Algorithm For each comparison, three distractors are generated from the baseline algorithm; three distractors are generated from the stated algorithm in this report. With the first comparison we will be able to see if the category information will help in selecting more suitable distractors. By comparing the results from the both evaluation, we will be able to see if semantic distance and category information will help improve the suitability of distractors.

4.3.2 Results

The evaluation contains 100 questions and is separated into 4 surveys, with each survey containing 25 questions. Each participant is free to choose one or more than one surveys. The purpose is to reduce the workload in each survey to get better responses. The surveys are sent to Year 1 students from School of Computing, National University of Singapore. There are 15 valid responses with each participant ranking each distractor with a different weight from 1 to 7. Half of the participants are native English speakers.

Each participant's rank will be the weight of the particular distractor in that question, i.e. if the user rank one distractor as rank "5", the weight of this distractor in this user's response will be 5. For each distractor of each question, the ranks of all users' responses are summed. As the more plausible the distractor is, the higher rank it will have, thus if the sum is higher, the approach is not as plausible as the other from user's point of view.

Table 3: Comparison 1 Baseline vs. Knowledge level 1 Algorithm

	Number of winning questions	Average score
Baseline	27	3.84
Level 1 Algorithm	23	4.10

Table 4: Comparison 2 Baseline vs. Knowledge level 3 Algorithm

	Number of winning questions	Average score
Baseline	21	4.16
Level 3 Algorithm	29	3.49

Table 3 and Table 4 showed the detailed result of each comparison. If for any question, the sum of weight from all participants for one approach is bigger than the other, then this approach is considered to have won this question. The "average score" is the average sum of weight from each approach for all questions. The lower the average score is, the better performance this approach has gained.

From Figure 2 we can see that in the first comparison, the baseline algorithm actually outscored the knowledge level 1 generation algorithm by 4 questions, with a sum of weight lower than 0.26. From Table 3 we can see that in the second comparison, the knowledge level 3 generation algorithm surpassed the baseline algorithm by 8 questions, with the average weight of 3.49 vs 4.16.

4.3.3 Analysis

In knowledge level 1 generation algorithm, there is no semantic distance calculation involved. If the target word to test has no strong category indication, for example, words like "venue", "week", it is possible that the knowledge level 1 algorithm will select some distractors that are not as plausible as those coming from the target word's synonym of synonym.

However, this problem is solved with the help of semantic distance calculator. In the knowledge level 3 generation algorithm, the distractors chosen are both semantic close and also category-related, which produced a relatively better experiment result.

Also in the baseline algorithm, it is possible that it will select words that are very rare in real life (Susanne, 2013), which may also have influence in the result.

5 Platform Viability and Usability Survey

We have thus far described and evaluated two critical components that can benefit from capturing the learner's news article context. In the larger context, we also need to check the viability of second language learning intertwined with news reading. In a requirements survey prior to the prototype development, two-thirds of the respondents indicated that although they have used language learning software, they use it infrequently (less than once per week), giving us motivation for our development.

Post-prototype, we conducted a summative survey to assess whether our prototype product satisfied the target niche, in terms of interest, usability and possible interference with normal reading activities. We gathered 16 respondens, 15 of which were between the ages of 18–24. 11 (the majority) also claimed native Chinese language proficiency.

The respondents felt that the extension platform was a viable language learning platform (BUG of 5; on a scale of 1 "disagreement" to 5 "agreement") and that they would like to try it when available forf their language pair (BUG of 5).

In our original prototype, we replaced the original English word with the Chinese translation. While most felt that replacing the original English with the Chinese translation would not hamper their reading, they still felt a bit uncomfortable (BUG of 5). This finding prompted us to review and change the default setting of the learning tooltip to simply add an underline to hint at the tooltip presence.

6 Conclusion

We have described `SystemA`, a software extension and server backend to transform the web browser into a second language learning platform. Leveraging web-based machine translation APIs and a static dictionary, it offers a viable user-driven language learning experience by pairing an improved, context-sensitive tooltip definition capability with the generation of context-sensitive multiple choice questions.

`SystemA` is potentially not confined to use in news websites; one respondent noted that they would like to use it on arbitrary websites, but currently we feel usable word sense disambiguation is difficult enough even in the restricted domain. We also note that respondents are more willing to use a mobile client for news reading, such that our future development work may be geared towards an independent mobile application, rather than a browser extension.

References

Hirst Alexander, Budanitsky; Graeme. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures.

Jordan L Boyd-Graber, David M Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In *EMNLP-CoNLL*, pages 1024–1033.

Klein Edward, Loper; Ewan. 2009. Natural language tool kit.

Dekang Lin. 1998. An information-theoretic definition of similarity.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, February.

Knoop; Sabrina Wilske Susanne. 2013. Wordgap - automatic generation of gap-filling vocabulary exercises for mobile learning.

Kristina Toutanova, H Tolga Ilhan, and Christopher D Manning. 2002. Extensions to hmm-based statistical word alignment models. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 87–94. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nicola Ueffing and Hermann Ney. 2003. Using pos information for statistical machine translation into morphologically rich languages. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 347–354. Association for Computational Linguistics.

Yorick Wilks and Mark Stevenson. 1998. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Nat. Lang. Eng.*, 4(2):135–143, June.

Kathryn Zickuhr, Lee Rainie, Kristen Purcell, Mary Madden, and Joanna Brenner. 2012. Younger americans' reading and library habits. *Pew Internet*.