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 (Rubin, 1975) as well as immersion into real scenarios (Naiman, 1978) 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 discov-

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

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

ered two key components that can be affected by this context modeling. We report on these developments here. In specific, we propose improved algorithms for two components: **(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 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 multiple choice translation test on the target word (Figure 1, right). Our learning platform thus can be viewed as three logical use cases: *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 $t = 3$ 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 importance to `SystemA` is the association of words to a news category, which is used downstream in both word sense disambiguation (Section 3) 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". However, note that Chinese news sites have a different categorization scheme, and as such, we first had to manually align the different categories based on our observation (See Table 1). After a survey of a number of English and Chinese 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. For Chinese, we additionally carried out Chinese word segmentation using the Stanford Chinese word segmenter (Chang et al., 2008). The remaining words are classified to a news category based on docu-

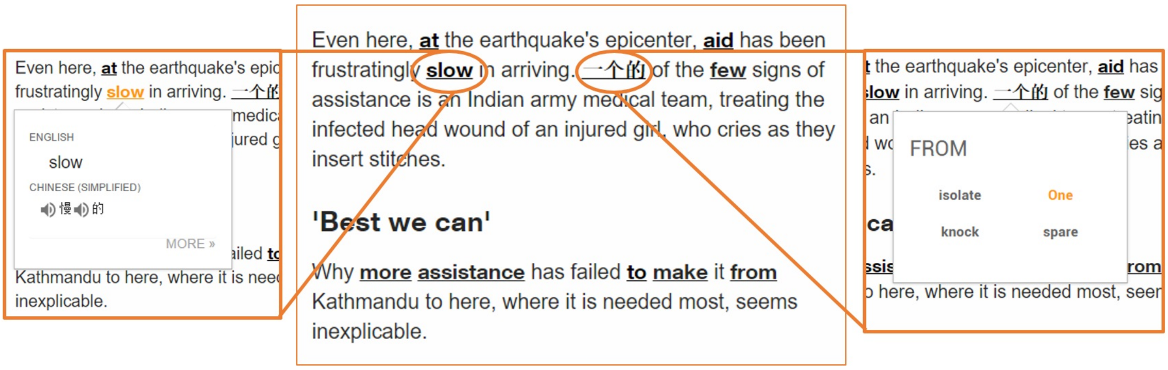


Figure 1: Merged screenshots of our Chrome extension on the CNN English article *Treacherous journey to epicenter of deadly Nepal earthquake*. Underlined components are clickable to yield tooltips of two different forms: (l) a definition for learning, (r) a multiple-choice interactive test.

Table 1: News category alignment between English and Chinese.

Chinese Category	English Category	Example Words
Entertainment	Entertainment	"design"
Military	World	"attacks"
Finance	Finance	"accounts"
Sports	Sports	"score"
International	World	"attacks"
Social	World	"attacks"
Technology	Tech	"phone"
Lady	Fashion	—
Auto	Travel	"natural"
Game	—	—
Education	—	—
—	Health	"stress"

ment frequency. A word w is classified to a category c if it appears a tunable threshold $\delta = 10$ more often than its average category document frequency. Note that a word can be categorized to multiple categories under this scheme.

3 Word Sense Disambiguation Component

Our extension needs to show the most appropriate translation sense based on the context. Such a translation selection task – cross-lingual word sense disambiguation – is a common problem in machine translation.

The contextual information that we use to assist WSD in our extension comes in two forms: the news category of the target word to be translated and the enclosing sentence.

3.1 Bilingual Dictionary and Baseline

`SystemA`'s server component includes a bilingual lexicon of English words with possible Chinese senses. The dictionary includes an indication of the relative frequency among Chinese senses, with their part-of-speech, per English word. The Chinese words in our bilingual lexicon are based from the publicly-available College English Test (CET 4) list, which has a breadth of about 4,000 words. Our baseline translation uses the most frequent sense: for an English word to be translated, choose the most frequent relative Chinese translation sense c from the possible set of senses C. This method has complete coverage, but as it lacks any context model, is the least accurate.

3.2 Approach 1: News Category

Topic information has been shown to be useful in WSD (Boyd-Graber et al., 2007). For example, consider the English word *interest*. In finance related articles, "interest" is more likely to carry the sense of "a share, right, or title in the ownership of property" ("利息" in Chinese), over other senses. Therefore, analysing the topic of the original article and selecting the translation with the same topic label might help disambiguate the word sense. For a target English e word, for each prospective Chinese sense $c \in C$, choose the first (in terms of relative frequency) sense that has the

Table 2: Example translations from our approaches to WSD. Target words are italicized and correct translations are bolded.

English Sentence	Dictionary	Baseline	POS	Machine Translation		
				Substring	Relax	Align
(1) ... a very *close* friend of ...	verb: 关闭, 合, 关 ... adj: 密切, ... 亲密 ...	关闭	密切	**亲密**	**亲密**	**亲密**
(2) ... kids can't *stop* singing ...	verb: 停止, 站, 阻止, 停 ...	**停止**	阻止	**停止**	**停止**	**停止**
(3) ... about Elsa being happy and *free* ...	adj: 免费, 自由, 游离, 畅, 空闲的...	免费	免费	**自由**	**自由**	**自由**
(4) ... why obama's *trip* to my homeland is meaningful ...	noun: 旅, 旅程 ... 旅游 ...	旅	旅	旅	**旅行**	**旅行**
(5) ... winning more points in the *match* ...	noun: 匹配, 比赛, 赛, 敌手, 对手, 火柴 ...	匹配	匹配	**比赛**	**比赛**	**比赛**
(6) ... *state* department spokeswoman Jen Psaki said that the allies ...	noun: 态, 国, 州, ... verb: 声明, 陈述, 述, 申明 ... 发言 ... adj: 国家的 ...	态	态	发言	发言人	**国家**

same news category as the containing webpage.

3.3 Approach 2: Part-of-Speech

Part-of-Speech (POS) are also useful for word sense disambiguation (Wilks and Stevenson, 1998) and machine translation (Toutanova et al., 2002; Ueffing and Ney, 2003). For example, the English word "book" can function as a verb or a noun, which gives rise to two different dominant senses: "reserve" ("预定" in Chinese) and "printed work" ("书"), respectively. As senses often correspond cross-lingually, knowledge of the English word's POS can assist disambiguation. We employ the Standford Log-linear Part-of-Speech tagger (Toutanova et al., 2003) to obtain the POS tag for the English word, whereas the POS tag for target Chinese senses are provided in our dictionary. In cases where multiple candidate Chinese translations fit the same sense, we again break ties using relative frequency of the prospective candidates.

3.4 Approaches 3–5: Machine Translation

Neighbouring words provide the necessary context to perform WSD in many contexts. In our work, we consider the sentence in which the target word appears as our context. We then acquire its translation from Microsoft Bing Translator[3] using its API. As we access the translation as a third party, the Chinese translation comes as-is, without the needed explicit word alignment to locate the target English word to translate in the original input sentence. We need to perform alignment of the Chinese and English sentences in order recover the target word's translation from the sentence translation.

Substring Match. As potential Chinese translations are available in our dictionary, a straightforward use of substring matching recovers a Chinese translation; *i.e.*, check whether the candidate Chinese translation is a substring of the Bing translation. If more than one candidate matches, we use the longest string match heuristic and pick the one with longest match as the final output. If none match, the system does not output a translation for the word.

Relaxed Match. The final rule in the substring match method unfortunately fires often, as the coverage of `SystemA`'s lexicon is limited. As we wish to offer correct translations that are not limited by our lexicon, we relax our substring condition, allowing the Bing translation to be a superset of a candidate translation in our dictionary

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

Table 3: WSD performance over our test set.

	Coverage	Accuracy
Baseline	**100%**	57.3%
News Category	2.0%	7.1%
POS	94.5%	55.2%
Bing – Substring	78.5%	79.8%
Bing – Relaxed	75.7%	80.9%
Bing – Align	76.9%	**97.4%**

(see Example 4 in Table 2, where the Bing translation “旅行” is allowed to be relaxed to match the dictionary “旅”). To this end, we must know the extent of the words in the translation. We first segment the obtained Bing translation with the Stanford Chinese Word Segmenter, and then use string matching to find a Chinese translation c. This technique significantly augments the translation range of our extension beyond the reach of our lexicon.

Word Alignment. The relaxed method runs into difficulties when the target English e's Chinese prospective translations (from our lexicon) may generate several possible matches.

Consider Example 6 in Table 2. The target English word “state” has corresponding Chinese entries “发言” and “发言”. Our relaxed approach yields “发言人” (“spokeswoman”, incorrect), because the Chinese translation “发言” (“state” as a verb).

To address this, we use the Bing Word Alignment API[4] to provide a possibly different prospective Chinese sense c. In this example, “state” matches “国家” (“country”, correct)from word alignment, and the final algorithm chooses “国家” as the output.

3.5 Evaluation

To evaluate the effectiveness of our proposed methods, we randomly sampled 707 words and their sentences from recent CNN[5] news articles, manually annotating the ground truth translation for each target English word. We report both the **coverage** (*i.e.*, the ability of the system is to return a translation) and **accuracy** (*i.e.*, whether the translation is contextually accurate).

Table 3 shows the experimental results for the six approaches. As expected, frequency-based baseline achieves 100% coverage, but a low accuracy (57.3%); POS also performs similarly . The category-based approach performs the worst, due to low coverage. This is because news category only provides a high-level context and many of the Chinese word senses do not have a strong topic tendency.

Of most promise is our use of web based translation related APIs. The three Bing methods iteratively improve the accuracy and have reasonable coverage. Among all the methods, the additional step of word alignment is the best in terms of accuracy (97.4%), significantly bettering the others. This validates previous work that sentence-level context is helpful in WSD.

4 Distractor Generation Component

Assesing mastery over vocabulary is the other key functionality of our prototype learning platform. The generation of the multiple choice selection test requires the selection of alternative choices aside from the correct answer of the target word. In this section, we investigate a way to automatically generate such choices (called “distractors”) (in English form) for a target word. We postulate “a set of suitable distractors” as: 1) having the same form as the target word, 2) fitting the sentence's context, and 3) having proper difficulty level according to user's level of mastery. As input to the distractor generation algorithm, we provide the target word, its part-of-speech (obtained by tagging the input sentence first) and the enclosing webpage's news category. We restrict the algorithm to produce distractors matching the input POS), and which match the news category of the page.

We can design the test to be more difficult by choosing distractors that are more similar to the target word. By varying the semantic distance, we can generate tests at varying difficulty levels. We quantify similarity by using the Lin distance (Lin, 1998) between two input candidate concepts in WordNet:

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

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

[5]http://edition.cnn.com/

where $p(c)$ denotes the probability of encountering concept c, and $lso(c1, c2)$ denotes the lowest common subsumer synset, which is the lowest node in the WordNet hierarchy that is a hypernym of both $c1$ and $c2$. This returns a score from 0 (completely dissimilar) to 1 (semantically equivalent).

If we use a target word e as the starting point, we can use WordNet to retrieve related words using WordNet relations (hypernyms/hyponyms, synonyms/antonyms) and determine their similarity using Lin distance.

We empirically set 0.1 as the similarity threshold – words that are deemed more similar than 0.1 are returned as possible distractors for our algorithm. We note that Lin distance often returns a score of 0 for many pairs and the threshold of 0.1 allows us to have a large set of distractors to choose from, while remaining fairly efficient in run-time distractor generation.

We discretize a learner's knowledge of the word based on their prior exposure to it. We then adopt a strategy to generate distractors for the input word learners based their level:

Easy: The learner has been exposed to the word at least $t = 3$ times. Two distractors are randomly selected from words that share the same news category as the target word e. The third distractor is generated using our algorithm.

Difficult: The learner has passed the Easy level test $x = 6$ times. All three distractors are generated from the same news category, using our algorithm.

4.1 Evaluation

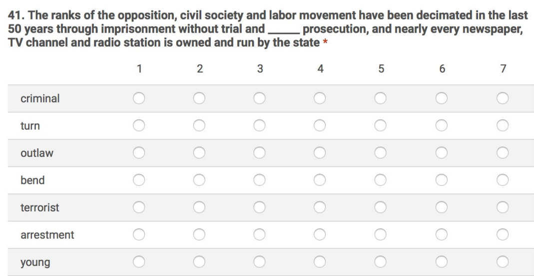
To compare with our proposed method, we reimplemented an existing distractor generation method used in WordGap system (Knoop and Wilske, 2013). WordGap adopts a knowledge-based approach: selecting the synonyms of synonyms (computed in WordNet) as distractors. That is, they select the most frequently used word, w1, from the target word's synonym set. Then they select the synonyms of w1 and call this set as s1. Synset s1 contains all the words that are synonyms of synonyms of the target word. Finally they select three most frequently used words from s1 as distractors. This we use is as our baseline approach for comparison.

Our proposed method adopts three different strategies to generate distractors according to user's knowledge level. In our evaluation, we study distractors generated for the two extreme cases, *i.e.*, knowledge level 1, and knowledge level 3. Therefore, we conduct a pairwise comparison – K1 vs. Baseline, and K3 vs. Baseline, using the same test dataset.

4.1.1 User Study

To compare the two approaches in generating distractors, we ask users to compare the plausibility of distractors. We randomly selected 50 sentences from recent news articles and then chose a noun or adjective from the sentence as the target word. In our survey, each question looks like a real MCQ quiz: we show the original sentence (leaving the target word as blank) as the context, and randomly display the six distractors and the target word as choices. Users are required to read the sentence and select the correct answer (that they think) as rating 1, and rank the other choices from 2 (most plausible) to 7 (least plausible) based on their plausibility. Figure 2 shows an example survey question.

We have two tests (K1 vs. Baseline, and K3 vs. Baseline) and each contains 50 questions. We further group 25 questions as one session, and give users the freedom to participate one or more sessions. Each question will be answered by at least five different users. Finally, we recruited 15 users from our university, and half of them are native English speakers. In average, each user participate two sessions.


41. The ranks of the opposition, civil society and labor movement have been decimated in the last 50 years through imprisonment without trial and ______ prosecution, and nearly every newspaper, TV channel and radio station is owned and run by the state *

	1	2	3	4	5	6	7
criminal	○	○	○	○	○	○	○
turn	○	○	○	○	○	○	○
outlaw	○	○	○	○	○	○	○
bend	○	○	○	○	○	○	○
terrorist	○	○	○	○	○	○	○
arrestment	○	○	○	○	○	○	○
young	○	○	○	○	○	○	○

Figure 2: A sample survey question

4.1.2 Results and Analysis

As each question is answered by five different users, we compute the average rating for each choice. A lower rating means a more plausible (harder) distractor. Unsurprisingly, the rating for all the target words is low (1.1 in average), as they are the ground truth. This implies that the users answered the survey questions seriously, and the evaluation quality is controlled. For each question, we determine a algorithm to be the winner if its three distractors as a whole (the sum of three average ratings) are more plausible than the distractors by another algorithm. We calculate the number of winning questions for each algorithm and compute the average score across the 50 questions. Winning more questions, and obtaining a lower average score denotes a better performance for an algorithm.

Table 4: Baseline vs. Knowledge Level 1

	Number of winning questions	Average score
Baseline	27	3.84
K1	23	4.10

Table 5: Baseline vs. Knowledge Level 3

	Number of winning questions	Average score
Baseline	21	4.16
K3	29	3.49

We display the results for Baseline vs. K1 and Baseline vs. K3, in Table 4 and Table 5, respectively. We see baseline outperforms the K1 algorithm by four more winning questions and 0.26 average score. Recall that, K1 algorithm is solely relied on category information, without taking word semantic relatedness into account. When a target word does not have a strong category tendency, *e.g.*, "venue" and "week", it is hard for K1 algorithm to select plausible distractors. On the other hand, we see context information (*i.e.*, new category) do play a key role, as K1 wins for 23 times.

In Table 5, we see our K3 algorithm significantly betters the baseline for both winning questions (8 more) and average score (0.67 lower). This further confirms that context and semantic information are complementary for distractor generation. As we mentioned before, a good distractor should fit the reading context and have a certain level of difficulty.

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 respondents, 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 (3.4 of 5; on a scale of 1 "disagreement" to 5 "agreement") and that they would like to try it when available for their language pair (3 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 (3.7 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

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

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 224–232, Stroudsburg, PA, USA. Association for Computational Linguistics.

Susanne Knoop and Sabrina Wilske. 2013. Wordgap-automatic generation of gap-filling vocabulary exercises for mobile learning. In *Proceedings of Second Workshop NLP Computer-Assisted Language Learning at NODALIDA*, pages 39–47.

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

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Neil Naiman. 1978. *The good language learner*, volume 4. Multilingual Matters.

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

Joan Rubin. 1975. What the" good language learner" can teach us. *TESOL quarterly*, pages 41–51.

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. *Natural Language Engineering*, 4(2):135–143, June.