Automatic Extraction of High-Quality Example Sentences for Word Learning Using a Determinantal Point Process

Arseny Tolmachev
Graduate School of Informatics
Kyoto University
Yoshida-honmachi, Sakyo-ku
Kyoto, 606-8501, Japan
arseny@nlp.ist.i.kyoto-u.ac.jp

Sadao Kurohashi
Graduate School of Informatics
Kyoto University
Yoshida-honmachi, Sakyo-ku
Kyoto, 606-8501, Japan
kuro@i.kyoto-u.ac.jp

Abstract

Flashcard systems are effective tools for learning words but have their limitations in teaching word usage. To overcome this problem, we propose a novel flashcard system that shows a new example sentence on each repetition. This extension requires high-quality example sentences, automatically extracted from a huge corpus. To do this, we use a Determinantal Point Process which scales well to large data and allows to naturally represent sentence similarity and quality as features. Our human evaluation experiment on Japanese language indicates that the proposed method successfully extracted high-quality example sentences.

1 Introduction

Learning vocabulary is a crucial step in learning foreign languages and it requires substantial time and effort. Word learning is often done using flashcards: a way of organizing information into question-answer pairs. An example of a flashcard for the Japanese word “柿” is shown on Figure 1 (a, b). Flashcard systems frequently use Spaced Repetition technique to optimize the learning process. The technique is based on the observation that people tend to remember things more effectively if they study in short periods spread over time (*spaced repetition practice*) opposed to *massed practice* (i.e. cramming) (Pavlik and Anderson, 2008; Cepeda et al., 2006). Anki[1] is one of the most well known open source Spaced Repetition System (SRS).

One major drawback of building a vocabulary with flashcards is that most of the time cards look like the one displayed on Figure 1 (top): flashcards often lack usage context information. A question card is usually *a word alone*, an answer card could contain a *fixed single* example sentence present. The example does not change from repetition to repetition, and as a result does not show the full spectrum of word usage. However, humans do not use isolated words for communicating. Words are always surrounded by other words, forming word usages. Learning these word usages is as important as learning words themselves.

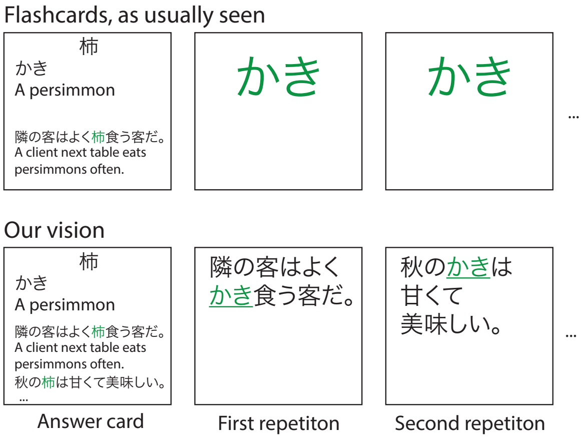

Figure 1: Flashcards for the word “柿”

To enhance the learning experience, we propose a novel framework of learning words using flashcards. Instead of showing only a single field like reading or writing of a flashcard in the question card similarly to the Figure 1 (top), we propose to use *example sentences* in both types of cards, see Figure 1 (bottom). Moreover, we want to show a *new* example sentence on each repetition as the question. This approach gives users an opportunity to learn correct word usages together with the words themselves. Obviously, implementing it requires a huge number of example sentences.

Because of this, we focus on automatic extraction of high-quality example sentences to be

[1] http://ankisrs.net

used in a flashcard system as questions. Collecting an enormous number of high-quality example sentences manually does not scale well. Words can have multiple senses and different usage patterns. A database containing dozens of sentences for each sense of each word would need to contain millions of different sentences. For a set of example sentences, we say that they are of high-quality if the sentences have the following properties.

- (Intrinsic) Value: Each individual example sentence should not be bad, for example ungrammatical, a fragment or unrelated to target word. Additionally, the sentences should not be too difficult for learners to understand them.

- Diversity: Inside a set, the sentences should cover different usage patterns, and word senses.

In addition we would like our method to support rare words and rare word senses.

For the task of example extraction, we are given a huge monolingual text corpora and a **target** word or a phrase to output a set of high-quality example sentences.

We propose a system architecture consisting of two components: a **search engine** which indexes a huge raw corpus and can produce a relatively high number of example sentence candidates, and the **selection part**, which takes the list of candidates and selects only a few of them. The search system is designed in a way so the selected sentences are syntactically rich near the target word (the target word has parents/children).

The DPP allows us to naturally represent data in terms of scalar quality and vector similarity. Additionally, the DPP has several interesting properties. For example, it is possible to compute a *marginal* probability of drawing a subset of items from a DPP efficiently. Marginal here means a probability of inclusion of a given set in *any subset drawn from the DPP*. Furthermore, it is proven that this marginal probability measure is submodular. Because of this, it is possible to build a greedy algorithm with reasonable guarantees, which selects items one by one, using the marginal probability measure as a weight. Also, the DPP is computationally and memory efficient. The computation of marginal probabilities can be performed linearly in respect to number of sentence candidates. This makes it possible to use the DPP with tens thousands of candidates in near-realtime scenarios.

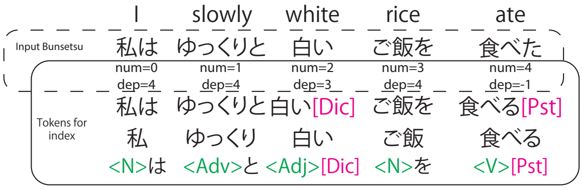

Figure 2: Word to token conversion for indexing a sentence. Tokens contain lexical information (black), POS tags (green) and conjugation forms (magenta). Dependency information is common for a set of tokens spawned from a single word. This information consists of word position and dependency position.

We have performed a human evaluation experiment which has shown that our method was preferred by Japanese learners and a teacher compared to two baselines.

2 Dependency Aware Search Engine

We want example sentences to have different possible usages of a target word. For example, verbs should have multiple arguments with different roles and in general it is better to have the vicinity of a target word syntactically rich. We use dependency information for approximating this information. For accessing syntactic information, we automatically tokenize raw text, extract lemmas, perform POS tagging and parse sentences into dependency trees.

To select syntactically rich sentences on a scale of a huge corpus, we have developed a distributed Apache Lucene-based search engine (Tolmachev et al., 2016) which allows to query not only on keywords as most systems do, but on dependency relations and grammatical information as well. We use this search engine to retrieve a relatively large set of example sentence candidates.

Search engines usually build a reverse index based on tokens, which are computed from the original document. We encode seed tokens for our engine as concatenation of lemma form and conjugation form tags, which are derived from the original text. For example, the verb 帰った (kaetta – “to leave” in past form) would be represented as “帰る+PAST”. Each token also stores the position of its parent.

The next step generates rewritten tokens from the seed tokens until no more new tokens can be

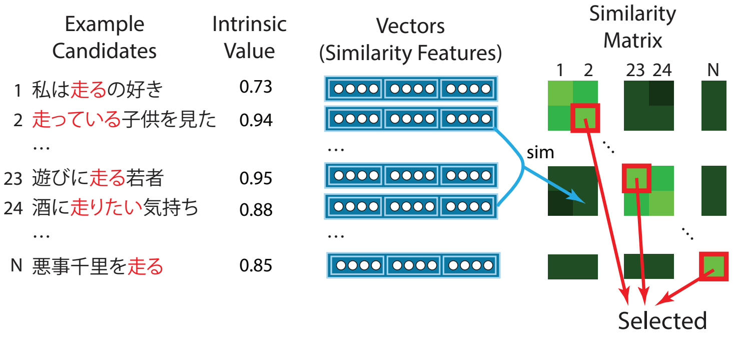

Figure 3: Example sentence selection. The objective is to select "best" and non-similar example sentences from the input list. Target word is marked red.

created using rewriting rules. Rewriting is done by replacing content word lexical information with part of speech information or removing some parts of tokens. For example, case markers of nouns are removed for some rules.

This representation allows to easily match same forms of different words while getting the benefits of reverse index in terms of performance. A list of created tokens for a raw sentence is shown in Figure 2. This example spawns three tokens for each of its word.

For selecting candidates we use queries which match a target word with up to 3 children or parents. The exact types of parents of children depend on POS of the target word. The number 3 was chosen to have balance with different arguments and to keep the syntactic vicinity of the target word diverse between the example sentence candidates.

3 Example sentence selection

After we have a relatively large list of example sentence candidates, we select a few of them as example sentences. The outline of the selection part is shown in Figure 3. In this section we describe the ideas behind the DPP and the way how we compute individual features.

3.1 Determinantal Point Process

In this section we provide a very basic explanation of the DPP inner workings. We invite interested readers to refer the original paper (Kulesza and Taskar, 2012) which gives a comprehensive overview of the DPP. In the supplementary material we show a toy task of greedily selecting a diverse subset of points from a plane to give an insight into how the DPP works.

Suppose we have a ground set $\mathcal{Y} = \{1...N\}$ of N items (in our case items are example sentence candidates from the search engine). In this stage we want to select a subset $Y \subseteq \mathcal{Y}$ s.t. $|Y| = k$. In its basic form, the DPP defines the probability of drawing a subset Y from a ground set as

$$\mathcal{P}_L(Y) \propto \det(L_Y) \tag{1}$$

Here L_Y denotes restriction of matrix L to the elements of Y, $L_Y = [L_{i,j}] : i, j \in Y$. L generally can be any semi-positive definite matrix, but for our task we compose it from two types of features: a **quality** scalar q_i and a **similarity** unit vector ϕ_i. Elements of L becomes a cosine similarity between the similarity features scaled by the quality features

$$L_{i,j} = q_i \phi_i^T \phi_j q_j. \tag{2}$$

The intuition behind the DPP as follows: because the right part of (1) contains determinant, when off-diagonal elements of L_Y get larger (meaning the cosine similarity of similarity features is large), then the determinant value, or in the other words, the probability of drawing Y, gets lower. At the same time, the DPP prefers elements with large values of quality features.

The DPP has a very interesting property. It is easy to compute **marginal** probabilities of inclusion of a set A in all subsets of the ground set $\mathcal{Y}$:

$$\mathcal{P}_L(A \subseteq \mathcal{Y}) = \frac{\sum_{Y:A\subseteq Y \subseteq \mathcal{Y}} \det(L_Y)}{\sum_{Y:Y\subseteq\mathcal{Y}} \det(L_Y)} = \det(K_A).$$

K_A is restriction of K with the elements of the set A (similar to (1)). K itself is called *marginal kernel* of the DPP and it can be computed as $K = L(L + I)^{-1}$, where I is an identity matrix.

Selecting diverse items

Because the elements of K can be used to compute the marginal probability of selecting a subset of items from the ground set, it is possible to use the marginal probabilities as a weight for a greedy selection algorithm.

In the beginning we have an empty set $A = \emptyset$. Then we repeatedly add an item i into the set A s.t. $i = \arg\max_i \det(K_{A \cup i})$ until the set A reaches the required size. Please note that this algorithm does not find a MAP answer, that problem is shown to be NP-complete.

Computational complexity

Dealing with L and K directly requires $O(N^3)$ floating point operations and $O(N^2)$ memory, which can be unwieldy for sufficiently large N.

Fortunately, if L is formulated as (2), it is possible to work around these requirements. Let B be a feature matrix with rows $B_i = q_i \phi_i$, so $L = B^T B$. Instead of computing $N \times N$ matrix L, we compute a $D \times D$ matrix $C = BB^T$. Note that if we have an eigendecomposition $L = \sum_{n=1}^{N} \lambda_n v_n v_n^T$, we can get the marginal kernel K by rescaling eigenvalues of L:

$$K = \sum_{n=1}^{N} \frac{\lambda_n}{\lambda_n + 1} v_n v_n^T.$$

Remember that non-zero eigenvalues of L and C are the same and their eigenvectors are related as well. Namely, the eigendecomposition of L is also

$$\left\{ \lambda_n, \frac{1}{\sqrt{\lambda_n}} B^T \hat{v}_n \right\}_{n=1}^{D},$$

where $\hat{v}_n$ are eigenvectors of C. Using this fact, we can compute the elements of marginal kernel K directly from the eigendecomposition of C and the feature matrix B:

$$K_{ij} = \sum_{n=1}^{D} \frac{(B_i^T \hat{v}_n)(B_j^T \hat{v}_n)}{\lambda_n + 1}.$$

Computation of a single element of K takes $O(D^2)$ floating point operations. For each step of the selection algorithm, we need to compute N new elements of K and compute N determinants of $|A| \times |A|$ size. In addition we need to compute an eigendecomposition of D. This leads to a total complexity of $O(D^3 + ND^2k + Nk^3)$ for selecting k items using the DPP, which is linear of N.

3.2 Similarity Features

We construct similarity feature vector as a weighted stacking of three individual feature parts

$$\phi_i = f([w_1 s_i^{\text{lex}} \, ; \, w_2 s_i^{\text{synt}} \, ; \, w_3 s_i^{\text{sema}} \, ; \, r])$$

and a parameter r which makes all sentences similar to each other, following the text summarization task in (Kulesza and Taskar, 2012). We set $r = 0.7$ in our experiments.

Three similarity feature parts are lexical, syntactic and semantic similarity. Feature weights w_i allow us to prioritize similarity feature components. Lexical and syntactic similarity features are created as count-based vectors and have large dimensionality. Transformation f here is a compression into a 600-dimensional vector using Gaussian random projections as recommended by Kulesza and Taskar (2012) to make the dimensionality of ϕ_i, D, small.

Lexical similarity features measure word overlap between two sentences, syntactic features measure structural (POS, grammar and dependency) similarity between two sentences and semantic features measure sense similarity of two sentences. Lexical similarity uses tf weighting inside example sentence candidate batch when inclusion of a content word is given a weight of 1.0; non-content words are given a weight of 0.1.

A **syntactic similarity** for two sentences should be higher if they have similar syntactic structure near the target word, meaning that it was used in a similar syntactic way. In other words, dependency structure, POS tags and grammatical words should be similar near the target word. For instance, let's consider sentences: “He is a fast runner”, “She is a slow runner” and “John isn't a good runner”. These three sentences have small content word overlap, but have exactly the same syntactic structure.

The idea for the syntactic similarity method is based on efficient calculation of graph similarity using graphlets. Graphlets are parts of graph, and it is shown by (Shervashidze et al., 2009) that they can be used for the fast approximate computation of graph similarity.

The main idea is to generate subtrees up to a certain size, by growing them from the target word and use those subtrees as features in the vector space. Overall, the syntactic similarity model can be thought of as a bag-of-subtrees model. Dependency trees in Japanese is build of *bunsetsu* – a unit which consist of a lemma with attached functional morphemes. Subtrees are treated as unordered because bunsetsu in Japanese can be moved on the same dependency level.

In the first step, the parse tree is *stripped* from lexical information for open parts of speech by replacing them with part of speech tags. Function words are left as they were.

Secondly, a set of bunsetsu subtrees up to size of 3 is generated from the stripped tree. The generation starts from the bunsetsu containing the target word and continues until no new subtrees can be created.

Finally, the feature space is expanded by deriving new subtrees. Bunsetsu can contain compound nouns like “参政権” (a right to vote) or “積み上げる” (to place on top of something) which are analyzed to consist of two lexical units. Grammatically, they are not much different from single unit words. This step ensures that sentences containing both several-unit and single-unit words are still going to be structurally similar.

A **semantic similarity** score should be higher if the target word is used in the same or a close sense. For computing semantic similarity from a context we use prototype projections (Tsubaki et al., 2013) on word2vec word representations (Mikolov et al., 2013).

Prototype projections assume that for triples of (A, relation, B) there exist prototypes in the form of frequently occurring and semantically related groups words at the end of each relation. For example, it is possible to run company, business or marathon. The computed representation makes it possible to distinguish between the distant senses. For a given triple (e.g run, object, marathon), you compute frequently occurring words of run and marathon over the same relation and compute SVD in each group. The top n right singular vectors in each end of the relation form a prototype subspace, and the original vector is projected into it.

For the actual feature we use a sum of prototype projections over all possible arguments of a target word. For instance, we use all present Japanese case relations if the target word is a verb, case relation and genitive case for nouns, and dependencies for adverbs and adjectives. For the each end of a relation use top 200 words to compute SVDs.

3.3 Quality Features

Quality features represent an *intrinsic value* of individual sentences as examples of word usage. Our quality feature is defined as a product of four components: $q_i = q_i^{\text{cse}} q_i^{\text{csy}} q_i^{\text{d}} q_i^{\text{g}}$.

Centrality

q_i^{cse} and q_i^{csy} are semantic and syntactic centrality, respectively. We want example sentences to be representative of usage patterns and meaning. Centrality captures that idea. It is computed using a respective similarity feature component (s_i^{synt} and s_i^{sema}) as a cosine similarity to a nearest centroid of a K-means++ clustering. We take $k = 30$ for semantic and $k = 10$ for syntactic centralities.

Relative difficulty

The next quality feature is relative difficulty. It is estimated from the difficulty of content words. Sentence difficulty d_s is computed from the word difficulty d_{w_i} using the formula

$$d_s = \left(\sum_{w_i \in s} d_{w_i}^4 \right)^{\frac{1}{4}}.$$

We used the fourth power to give the sum a light softmax effect: smaller values should have less effect on the final result, but the sentence length should still be a certain factor in the difficulty score. Word difficulties are estimated using web corpus word frequencies and Japanese Language Proficiency Test (JLPT) word lists.

Frequency component of word difficulty is computed as $d_w^{\text{freq}} = \lfloor \log_2(1 + w_f/500) \rfloor$. Words which should be known for JLPT N5 were given the difficulty $d_w^{\text{JLPT}} = 1$, words for N1 were assigned $d_w^{\text{JLPT}} = 5$ respectively with other values in between. The final word difficulty score is computed as $d_w = \min(d_w^{\text{freq}}, d_w^{\text{JLPT}})$.

Sentence difficulty is then converted into the quality feature component using a piecewise linear function $q_i^{\text{d}} = T(d_s + \text{bias}_d)$, which is defined as $T = [0, 0.6, 1, 0.9, 0.7, 0.6, 0.2, 0]$ at $[-\infty, -1, 0, 3, 5, 6, 8, \infty]$. The function is rather adhoc. It has a maximum of 1 at 0 and decreases to the left and right. We wanted to have positive and negative parts to decrease with the different rate. A bias value bias_d can shift the area of acceptable difficulties for a learner. For example, a bias value of $\text{bias}_d = -3$ would make the quality to be near 1 for the sentences which have the words with the difficulty at most for JLPT N3.

Goodness

The last part is goodness feature q_i^{g} which is 1 by default and assigns a low score to garbage sentences which are present in the web corpus. It also assigns low score to sentence fragments (some sentences from raw corpus start with case particles which in Japanese always comes after a noun) or clearly sentences which are useless for example sentences, for instance ones that contain random digits or alphabet.

4 Related Work

There exist human-curated databases of example sentences. Dictionaries contain example sentences which explain word usage, but usually those are fragments and not full sentences. Also, dictionary content usually has copyright restrictions. The Tatoeba Project[2] is a wiki-style database of example sentences maintained by human volunteers under open license. However, most of the sentences focus on relatively easy words and many of the sentences are very similar to each other.

Automated extraction of example sentences from a corpora has also been proposed. GDEX (Kilgarriff et al., 2008) describes semi-automated example extraction. The objective is to select example sentences for English learners and define a suitable example sentence as: (a) typical, showing frequent and dispersed patterns of usage, (b) informative, helping to educate the definition, (c) readable, meaning intelligible to learners, avoiding difficult words, anaphora and other structures that makes it difficult to understand a sentence without access to wider context. Sentence length, word frequency, information about the presence of pronouns and some other heuristics were used to judge the quality of sentences. Subsequently, the final example sentences for the dictionary were manually selected by editors.

There are numerous works which approach the problem of selecting example sentences mostly as a word sense disambiguation (WSD) problem (de Melo and Weikum, 2009; Shinnou and Sasaki, 2008; Kathuria and Shirai, 2012). Specifically, de Melo and Weikum (2009) proposed the use of parallel corpora to extract disambiguated sentences from an aligned subtitle database. One more important feature of that work is a concern about *diversity* of example sentences. They generate a set of 1,2,3-grams for each example sentence and use them for scoring example sentences, setting to zero scores for n-gram for the selected sentences. This approach used aligned corpora for WSD, which usually are small or belong to a specific domain, whereas example sentences should be from different domains and cover rare words. Also, the work does not consider sentence difficulty. In the evaluation by language learners we found out that sentence difficulty is a major factor for example sentence quality.

Kathuria and Shirai (2012) explore the use of disambiguated example sentences in a reading assistant system for Japanese learners. They create a system that assists reading by showing disambiguated example sentences that have the same sense as the word in the text.

Huang et al. (2016) have used neural network models to show example sentences which would help disambiguate close synonyms. However, this work does not try to extract globally diverse example sentences which cover the usage of a target word.

The DPP itself (Kulesza and Taskar, 2012) was used for document summarization by selecting sentences from a text and showing a diverse image search result tasks. We use several tricks from the former application.

5 Evaluation

Evaluating the suitability of example sentences for learning a foreign language is difficult. Firstly, it is not possible to assess the diversity of a sentence set when showing them to evaluators one by one. Also, the automatic evaluation of example sentences is possible if the problem is formulated such that the only criterion is that example sentences should be present for every sense of a word. However, such evaluation does not determine whether the example sentences are actually useful for learners.

5.1 Experiment Setup

We perform an evaluation experiment with Japanese language learners and a native teacher with two distinct main goals: to assess the performance of the example extraction system and to validate the assumptions on the meaning of the "quality" of example sentences. We use a web corpus with 0.8B sentences lexically analyzed by JUMAN and parsed by KNP.

The first goal is achieved by having participants vote on lists of example sentences and select their preferred lists. We deliberately use lists for the evaluation instead of showing single examples to make the spectrum of possible example sentences visible for each method. Showing sentences one

[2] http://tatoeba.org/eng/

by one would make it difficult to compare the diversity of different lists.

For the second goal, the evaluation was performed in the form of an interview. Participants were asked why they have or have not chosen specific lists of example sentences after the initial preference selection.

Three methods were used in the evaluation: the proposed one and two baselines. The proposed method is labeled **DPP** in the evaluation results. We have used a difficulty bias value $\text{bias}_d = -3$ to make the sentence difficulty appropriate for the learners around JLPT N3 level.

The first baseline was a method by de Melo and Weikum (2009). However, because our setting uses only monolingual corpora, only lexical centrality and diversity parts were used from this method. The method received the same set of example sentences as the DPP, namely search results biased towards syntactically rich sentences near a target word. The method is referred as **DeMelo**.

The second baseline was a simple uniform random sampling without replacement. The data, again, was a list of example sentence candidates from the search system, not raw examples. This method is referred as **Rand**.

For the experiment we have used 14 Japanese words. Each chosen word has more than one sense and different usages. Words were also chosen to be relatively easy, to be likely familiar to language learners of lower intermediate level.

For each of the words, top 10k search results from the search engine were extracted as example sentence candidates. Each of the words had more than 10k containing sentences. After that, 12 sentences were extracted by each method from each list. That yields a total of 14 × 12 × 3 sentences which were presented to participants of the experiment.

The first part of the evaluation experiment used Japanese language learners as participants. For each word, participants were presented three lists of example sentences produced by three methods. The lists were placed side by side in a random order to force participants to read sentence lists in a different order every time. Participants were asked to select a list which was more useful from their point for putting sentences on flashcards. After a participant would select a personally preferable list, anonymized names for methods were displayed and the participant was asked to explain the reasons behind the selection.

#	FC	Level	Rand	DeMelo	DPP
1		N1	**7**	4	3
2		N1	**8**	0	6
3		N1	4	**7**	3
4	*	N1	2	3	**9**
5	*	N1	3	2	**9**
6	*	N2	5	3	**6**
7		N2	4	**6**	4
8		N2	5	2	**7**
9	*	N2	3	4	**7**
10	*	N3	0	1	**13**
11	*	N4	3	1	**10**
Total			44	33	77
Percentage			29%	21%	50%

Table 1: Learners' votes on the best example lists. Bold numbers are the majority for a person. FC means the experience of using flashcards. Level is approximate JLPT-style Japanese language proficiency from N5 (lowest) to N1 (highest).

The second part experiment was performed by showing the same example sentence lists to a native Japanese language teacher. In addition to selecting the best list, a teacher was asked to rank from 1 to 5 how appropriate the list was for students of approximately N3 and N2 JLPT levels. N3 is similar to intermediate and N2 to upper-intermediate levels in English. Similarly to the learners' case, no explicit criteria were given. Unfortunately, because of time limitations only one teacher have participated in the second part of the evaluation.

5.2 Results

The first part of the evaluation was performed with 11 learners. The evaluation took about 1.5 hours per learner in average. Vote counts for users and aggregated counts are shown in the Table 1. DPP got about a half of all votes, which is a positive aspect of the proposed method. It also got a majority for every participant who had the experience of using flashcards or spaced repetition systems. This shows that these example sentences are going to be useful inside the flashcards.

For the initial selection, the teacher commented that the best list was selected as if examples were for learners of N3 level. The votes on the initial selection were 0, 4, 10 for Rand, DeMelo and DPP respectively. Average lists ranks were 3.36, 3.79, 4.64 for N3 and 3.86, 4.21 and 4.36 for N2 learner levels.

Evaluation by the teacher assigns the DPP system as the best for N3 learners both by votes and by average rank. For N2 learners a score for DPP was lower, at the same time the score for DeMelo has raised. Score for Rand was the lowest.

The teacher explained the reason for selection as the following. Non-target words in a sentence should not be too difficult. A sentence should not depend on outer context like as if it was inside the conversation or about current affairs. The sentences should be short and the usages of the target words should be common. This criteria are strongly aligned with the objectives DPP uses for sentence extraction, which seems to be the reason for its high appraisal by the teacher.

If examples would be selected for N2-like learners, a sentence should include more diverse structures and usage. However, some high-level students had a different point of view.

There were cases when learners discarded a list because of a sentence they did not like or selected a list because of a sentence they liked very much. We tried to analyze the patterns of such sentences with a possibility for the further improvement of example sentence extraction.

6 Discussion

During the evaluation experiment, participants were asked to explain their choices about lists and criteria they were using.

Generally, list diversity was regarded as one of the main criteria for the selection. Semantic and lexical diversity was the mainly referred part. However, grammatical diversity was named as well. By grammatical diversity participants meant, usually, usage of words in different grammatical forms. Other themes that frequently came into criteria for the selection were sentence difficulty and *how interesting* were the sentences. Each of the points is discussed in greater detail below.

Diversity Diversity was the main idea behind the work for the present study and it was validated by answers of the participants. Most of them have stated that non-similarity of a sentence list was one of the main criteria for the selection.

All three used methods were specialized to produce non-similar sentences. DeMelo explicitly tries to select sentences with frequent words and penalize such words in next selections. Diversity of sentences using random sampling depends on the distribution in the candidate set.

For DPP, features were explicitly crafted to deal with semantic and syntactic similarity in addition to lexical similarity. Based on the results, there were cases where DPP was better in terms of diversity and the cases when it was worse.

One example of good performance in this regard was the word “卵” (an egg). In addition to the usual meaning of an egg in a sentence like “それには多くの卵を割る必要があります” (You would need to break a lot of eggs to make that), DPP also displayed several sentences for the usage like “医師の卵に期待が集まっている” (There are a lot of expectations in the future doctors) with the meaning of “future profession”. Other methods did not produce example sentences with this sense.

A similar, but mixed result is sentences for the word “頭” (a head). DPP selected 6 sentences that have the regular meaning of the word as “head” like “彼女は僕の頭に手をかける” (She puts a hand on my head). However, the other 6 had the meaning of beginning of a time period like in the sentence “今年の頭に撮った写真です” ([This is] a photo I’ve taken in the beginning of this year).

Difficulty Sentence difficulty was also one criterion experiment participants used for selecting lists. The initial assumption for the creation of the system is that example sentences should be easy to understand and as short as possible. We designed an algorithm which selects example sentences for flashcard questions and thought that it was good to minimize question reading time.

The feedback of participants on this topic was divided. Learners of lower proficiency levels have agreed with our vision, while learners of higher proficiency levels have shown preference for more difficult example sentences. For the last user group, there were several opinions that example sentences selected by DPP were plain as if they come from a textbook. In comparison to that such learners preferred, more difficult, natural (in contrast to artificially created examples), and interesting example sentences.

We believe that this effect can be explained with learners’ familiarity with the target word of example sentence. If a learner is not familiar with the target word, then the other words are expected to serve mostly as explanation for the target’s meaning and the sentence itself should be easier. If a learner is generally familiar with the word, that context given by an example sentence helps learner to learn and remember usage situations of the target. Sentences in this period of the familiarity could be harder.

It seems that we should talk not about good example sentences *in general*, but about good example sentences *for a learner at some point in a learning process*. Static example lists are not going to solve this problem efficiently, but an educational tool like an SRS can. It has access not only to learner's general knowledge level, but for the learning process data for individual words as well. Using this information about learners, an example extraction system can provide the best examples learner needs at that point of time.

Interestingness Another criteria that was used by learners for selecting sentences was if the sentences were *interesting*. During the evaluation, there were the cases when the choice between lists was made on a single interesting sentence, disregarding the fact that the list have contained mostly inferior and low-quality sentences like complete fragments. There were 3 main types of such sentences.

The first type had sentences, interesting or unusual for a certain participant. We could not generalize this category further.

The second type was sentences having a story. For example, "画像が汚いのは、携帯カメラで撮ったからです、今度綺麗な写真でも撮っておきましょう" (Image quality is bad because it was taken by a mobile phone. Let's take a good picture next time.) vs "画像が汚かったりしたら買う気しませんからね" (I don't want to buy it since the image quality is bad). These two sentences have the same word usage of "dirty" (image is dirty = image quality is bad). However the first one has a cause-effect relation and was more liked because of that.

The third type as sentences displaying a vivid image. For instance, "旧ソ連の宇宙飛行士ガガーリンの有人宇宙飛行「地球は青かった」"(A famous Soviet astronaut Gagarin have said: "The Earth is blue").

Interesting content usually occurs only in relatively lengthy sentences containing many different words. Because of the conservative difficulty settings we used for the experiment, the DPP method was heavily biased against such sentences. Interestingness is difficult to define and measure, but we believe that it is worth investigating in the future.

7 Conclusion and Future Work

We have implemented an example extraction system for usage in a flashcard system for Japanese language learners. It uses Determinantal Point Process — a method for modeling diverse datasets as a framework which allows to select non-similar and high quality sentences at the same time.

While the example extraction system is developed for Japanese, but the underlying methods have little Japanese specific parts. The system itself is unsupervised and has only a tokenizer, morphologic analyzer and dependency parser as software dependencies. All other data can be created from a raw corpus analyzed by these three tools.

Experiments have shown that the proposed DPP-based method is useful for extracting example sentences. However the content and difficulty of example sentences are a non-trivial problem and it would be promising to consider ways to further improve the content and quality of example sentences. We also want to perform evaluation experiments using an actual SRS (Tolmachev and Kurohashi, 2017).

Acknowledgments

The first author thanks Japanese government and the Ministry of Education, Culture, Sports, Science and Technology for the MEXT scholarship which made this work possible. We thank Yugo Murawaki and Raj Dabre for proofreading and very helpful comments. We would also like to thank anonymous reviewers for their feedback.

References

Nicholas J. Cepeda, Harold Pashler, Edward Vul, John T. Wixted, and Doug Rohrer. 2006. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3):354–380.

Chieh-Yang Huang, Nicole Peinelt, and Lun-Wei Ku. 2016. Automatically suggesting example sentences of near-synonyms for language learners. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 302–306. The COLING 2016 Organizing Committee.

Pulkit Kathuria and Kiyoaki Shirai. 2012. Word Sense Disambiguation Based on Example Sentences in Dictionary and Automatically Acquired from Parallel Corpus. In *Advances in Natural Language Processing*, number 7614 in Lecture Notes in Computer Science, pages 210–221.

Adam Kilgarriff, Milo Husk, Katy McAdam, Michael Rundell, and Pavel Rychl. 2008. GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the 13th EURALEX International Congress*, pages 425–432, Barcelona, Spain. Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra.

Alex Kulesza and Ben Taskar. 2012. Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286.

Gerard de Melo and Gerhard Weikum. 2009. Extracting sense-disambiguated example sentences from parallel corpora. In *Proceedings of the 1st Workshop on Definition Extraction*, WDE '09, pages 40–46, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

Philip I. Pavlik and John R. Anderson. 2008. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology. Applied*, 14(2):101–117.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 488–495. PMLR.

Hiroyuki Shinnou and Minoru Sasaki. 2008. Division of Example Sentences Based on the Meaning of a Target Word Using Semi-Supervised Clustering. In *LREC 2008*.

Arseny Tolmachev and Sadao Kurohashi. 2017. Kotonoha: An example sentence based spaced repetition system. In *Proceedings of the Twenty-third Annual Meeting of the Association for Natural Language Processing*, pages 847–850, Tsukuba, Japan.

Arseny Tolmachev, Hajime Morita, and Sadao Kurohashi. 2016. A grammar and dependency aware search system for japanese sentences. In *Proceedings of the Twenty-second Annual Meeting of the Association for Natural Language Processing*, pages 593–596, Sendai, Japan.

Masashi Tsubaki, Kevin Duh, Masashi Shimbo, and Yuji Matsumoto. 2013. Modeling and learning semantic co-compositionality through prototype projections and neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 130–140, Seattle, Washington, USA. Association for Computational Linguistics.