EECS 487 Final Report

Maestro: Topic-aware, Intelligent Flashcards

Casper Guo Nick Lovell

December 2023

Contents

List of Tables

List of Figures

1 Overview & Motivation

Vocabulary learning is a fundamental part of learning any foreign language. In a traditional classroom setting, this is often accomplished by studying a list of vocabulary provided by the instructor or the textbook. Students also frequently use flashcard apps to supplement their vocabulary learning. In this project, we implement an intelligent flashcard app that improves this process in two primary ways.

First, we observe that learners will stay more motivated and engaged when they are interacting with materials that are of specific interest to them. We design *Maestro* to work with a user-supplied corpus and use automatic flashcard creation and curation to generate the study materials, thus reducing user burden.

Second, basic flashcard apps surface user-created flashcards in random orders without considering users' previous interactions, the relationship between cards, and the relative importance of cards. *Maestro* implements a spaced repetition system (SRS) that sorts cards into different review time bins based on student feedback (see SRS methodology section). Within each review bin, we additionally sort the cards based on a combination of language importance score, corpus importance score, and sentence quality metrics. This ensures the cards first surfaced to users are optimized for enhancing long-term recall performance and accelerating progress towards competency in the corpus specifically and in the foreign language generally.

2 Interface

Please refer to the Figures section to see a mock interface of the Maestro language learning app in Figures 6, 7 and 8 for login splash, flashcard interface, and end-of-review splash, respectively.

3 Related Work

Learning with a SRS stands in contrast with massed learning (cramming). On a conceptual level, SRS is based on the intuition that reinforcement interval (the amount of time between showing the learner the same information) needs to be shorter for content that the learner is less familiar with, and vice versa. A recent meta-analysis [1] found that spaced learning consistently shows benefits in retention compared to massed learning. The studies included in this meta-analysis tested subjects' retention at a large variety of post-learning durations, ranging from one minute to eight years. One included study deals directly with learning vocabulary in a foreign language [2]. In this study, 4 subjects learnt 300 English-foreign language word pairs with the learning sessions spaced at intervals of 14, 28, or 56 days. When tested five years later, the subjects could recall 36 to 60 percent of the learned pairs. This study demonstrates the long-term benefit of SRS in a language-learning context.

Another area of prior work that we took inspiration from is the automatic generation of question-and-answer pairs from a corpus. Although our flashcards are directly sourced from a parallel corpus and no complicated algorithm is needed, we modelled much of our ranking metrics after the work of Tolmachev, Kurohashi, and Kawahara [3].

The architecture they implemented selects sentences from a large Japanese corpus and seeks to present to the learners different senses of Japanese words in context. This is done by first computing the lexical, syntactic, and semantic similarity between sentences. Then all the sentence-sentence similarity pairs can be represented as a matrix and determinal point process is used to extract a set of k sentences that maximizes the diversity (minimizes similarity). This set of sentences is then ranked by the product of three metrics.

1. Semantic and syntactic centrality measures how representative the sentence is for common usage patterns. This metric is calculated as the distance to the nearest centroid after applying the K-Means++ clustering algorithm to the similarity matrix.

2. Relative difficulty assesses the overall comprehensibility of the sentence for non-native speakers. This metric is primarily based on in-corpus word frequency, with higher-frequency words being assigned a lower difficulty score. To adapt to the typical learning pattern of a non-native speaker, this metric is adjusted based on JLPT (Japanese Language Proficiency Test) vocabulary lists. The rationale is that some words commonly taught to

beginners are, in fact, not frequently used by native speakers. *Maestro* implements a similar metric, which we term the language importance score.

3. Sentence goodness scores are based on a set of heuristics for evaluating the sentences' usefulness for language learning. For example, sentences that contain high percentages of punctuations or numbers are not semantically rich and therefore are of little interest to language learners. For *Maestro*, we use a sentence quality metric based on sentence length.

This sentence retrieval and ranking method is evaluated against random retrieval and a search engine-like retrieval system. The set of top sentences is shown to 23 learners and one instructor, with most participants preferring the sentences retrieved by the authors' architecture.

Maestro improves on this architecture in the following ways:

1. Since one primary motivation for *Maestro* is to enable users to work with their chosen corpus instead of a general corpus, we want to also evaluate tokens for their importance in the source corpus. To this end, we used topic modelling techniques to infer the importance of each token to the overall meaning and themes of the corpus.

2. Combining SRS with other sentence ranking metrics leads to sentences being shown in a more coherent order. In Tolmachev et al.'s infrastructure, the non-target words in a sentence only contribute to the ranking via the frequency-based relative difficulty score. Whereas with the aid of SRS, this difficulty score can be fine-tuned based on prior user feedback. It wouldn't make sense to surface to the user a sentence comprising mostly of high-frequency yet never-before-seen words.

4 Methodology

Maestro can, in principle, ingest a monolingual corpus and construct the parallel corpus using machine translation. However, for ease of implementation in this project, we use a high-quality, readily available parallel corpus.

4.1 Dataset

We use the Spanish-English parallel corpus provided by the OpenSubtitles dataset [4]. These corpora are provided in webvtt format, which is a standard for storing timed text. Here is a typical record in the corpus:

```
15
00:03:09.689 --> 00:03:12.692
position:50.00%,middle align:middle size:80.00% line:79.33%
and your version of an answer
was to turn into a wind-up monkey.
```

In this format, each subtitle entry includes a sequence number, a timestamp indicating when the subtitle appears and disappears on screen, and the subtitle text itself.

We need to perform an alignment process to create a parallel corpora for the purpose of our system, as there is not always a 1:1 mapping due to differences in languages, inconsistencies in audio dubbing, and various other reasons. This process involves comparing the duration and sequence and checking for overlap, which can be done trivially and efficiently.

Subtitle alignments are both provided by OpenSubtitles and can also be created with various open-source scripts.

The files also contain other metadata, such as the position and alignment of the subtitles. We do not make use of these metadata in our project.

4.2 Data Processing

We use the *webvtt-py* library to extract all caption texts from the vtt files. We then use a SpaCy pre-trained pipeline [5], specifically the *es_core_news_sm* pipeline to preprocess the captions.

The preprocessing consists of the following steps:

1. The caption is put into lowercase.
2. Non-word characters, such as the newline character, are removed.
3. SpaCy's built-in tokenizer is used to tokenize each caption, discarding all punctuations.
4. For each token, we store its lemma as calculated by SpaCy's lemmatizer.

Notably, stop words are not removed from the corpus as they are significant in a language-learning context and we want to avoid destructive modifications to the texts.

4.3 Language Importance

For each token, a language importance score is calculated. This score is based on word frequency in the CERA corpus, a large Spanish reference corpus, and the typical vocabulary sizes of a foreign language learner at each CEFR (Common European Framework of Reference for Languages) level [6] [7]. The score ranges from 0 to 1 according to table 1. Since Spanish is morphologically rich, we carry out the word frequency lookup using the lemmatized form of each token.

Intuitively, this score answers the question "how important is knowing this token for learning the langauge in general".

4.4 Corpus Importance & Topic Modelling

For each token, we also calculate a corpus importance score using topic modelling methods. Intuitively, this score answers the question "how important is knowing this token for understanding the chosen corpus specifically".

4.4.1 Term Frequency-Inverse Document Frequency (tf-idf)

Since the corpus importance score can only be meaningfully defined within the context of each film, we define the entire set of captions belonging to a film as the corpus and each subtitle entry in the vtt file as a document. We calculate idf with add-one smoothing and we use sublinear tf to reduce the significance of high-frequency stop words.

$$\text{idf} = \log \frac{\#\text{docs} + 1}{\text{df} + 1} + 1 \tag{1}$$

$$\text{tf} = 1 + \log \text{tf} \tag{2}$$

Lastly, we collect all the tf-idf scores and apply min-max normalization.

4.4.2 Latent Semantic Analysis

Latent semantic analysis (LSA) builds upon the previously computed tf-idf feature matrix and performs singular value decomposition (SVD) to extract latent topics. For each topic, the top x words can be extracted but LSA doesn't generate any numerical weights for each token. This makes extracting the per-token corpus significance score with LSA difficult since an ordinal to continuous conversion must be performed. Despite this inherent obstacle, LSA remains a useful benchmark for evaluating other models. Our LSA implementation is based on the scikit-learn package [8].

4.4.3 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is also derived from the tf-idf feature matrix. It is a Bayesian method that iteratively optimizes the per-document and per-word topic distributions. For each token, LDA can produce its weight in each topic, and the highest of these weights can serve as a proxy for the corpus importance score. our LDA implementation is based on the gensim package [9].

4.5 Sentence Quality Metric

We want to prioritize sentences of intermediate lengths. Long sentences are challenging for beginners while short sentences are often not semantically rich and don't properly demonstrate the grammatical structures typical of the foreign language.

For a desired sentence length l^* and the actual length l, a sentence quality metric can be calculated with a heuristic formula such as equation 3. Alternatively, a piecewise function may be defined by hand.

$$1 - 0.1 \times |l - l^*| \tag{3}$$

4.6 Integration

In order to calculate per-sentence scores, we must first combine the per-token scores.

We use equation 4 to combine language importance scores. This equation allows a sentence to accumulate importance over many words, even if each of them are relatively lower-frequency. This equation then serves as a further incentive for longer sentences.

$$L_s = (\sum_{w_i \in s} L_{w_i}^4)^{\frac{1}{4}} \tag{4}$$

where L_s is the sentence language importance score and L_{w_i} is the token language importance score.

Since our implementation of tf-idf allows for zeros and the weights returned by LDA can be very small, the traditional method of combining tf-idf, namely summing the logs, is not suitable. Instead, we take advantage of the min-max normalization that we've performed and simply take the average of the token corpus importance scores to compute the sentence's corpus importance score, as shown in equation 5.

$$C_s = \frac{\sum_{w_i \in s} C_{w_i}}{|s|} \tag{5}$$

where C_s is the sentence corpus importance score and C_{w_i} is the token corpus importance score.

The final sentence score is then calculated as a weighted sum of the sentence language importance score, the sentence corpus importance score, and the sentence quality metric. If tf-idf is used to compute the corpus importance score, then the three inputs are evenly weighted.

4.7 Spaced Repetition System

4.7.1 User Interface

The user workflow is set up as follows. See figure 1 for a schematic of our implementation of SRS.

1. Present a flashcard to the user, which may be new, or in a review process.
2. User clicks on tokens they are **unfamiliar** with.
3. Adjust the token difficulties and calculates the new card difficulty and interval.
4. Determines the next review date (or, if to be reviewed again in the same day, in what order).
5. Selects a the next card after the heuristic calculates new information about the cards.

In general, script updates the difficulty of each token based on user feedback (known vs. unknown tokens), which in turn influences the difficulty of the flashcard and its subsequent review intervals. The system differentiates between new cards and cards that are in the learning phase, adjusting the learning process accordingly. If a user struggles significantly with a card (indicated by a high unknown token ratio), the card is scheduled for same-day re-review rather than being pushed further into the future.

4.7.2 Algorithm

Each token in a sentence has an associated difficulty level, which is determined based on the corpus statistics: This initial difficulty is set based on the token's importance in the corpus, with a default value of 0.5 if not specified in the corpus stats. This is set when a user creates their study deck for the first time.

After each review, the difficulty of each token is updated based on user feedback:

```
if known:
    difficulty *= (1 + known_difficulty_adjustment)
else:
    difficulty *= (1 + unknown_difficulty_adjustment)
```

The difficulty for known tokens is decreased (e.g., by 10%), and for unknown tokens, it is increased (e.g., by 20%).

4.7.3 Review Interval Calculation

The interval for the next review of a flashcard is determined based on the current interval, token familiarity, and the sentence score:

Base Interval Adjustment:

```
if high_ratio_of_unknown_tokens:
    base_interval = current_interval * 0.5
else:
    base_interval = current_interval * 2
```

Adjusted Interval:

$$\text{New Interval} = \text{Base Interval} \times (1 + \gamma \times \text{Token Familiarity} + \delta \times \text{Sentence Score}) \tag{6}$$

Where γ and δ are weights for token familiarity and sentence score, respectively, and are hyperparameters that can be easily manipulated. The token familiarity is the average of the individual token scores.

The overall difficulty of a card is the average difficulty of all its tokens:

4.7.4 Selecting the Next Flashcard for Review

The next flashcard to be reviewed is based on two main factors: the due date of the flashcard and its educational value, which is calculated using sentence score and token familiarity.

1. First, the system retrieves all flashcards that are due for review based on the user's review schedule. This includes both cards that are up for regular review and new cards that haven't been reviewed yet.

2. The due flashcards are sorted based on a combination of their sentence score and token familiarity.

3. The system then picks the flashcard with the highest priority based on this sorting for the user to review next.

5 Experiments

Our experiments are centered around the effectiveness of the different topic modelling methods because the corpus importance score is the most tunable part of our project. Having high-quality, high-relevance retrieval capabilities means we can surface more relevant flashcards to the learner.

5.1 Evaluation Method

We base our evaluation method on the observation that words which are highly relevant to the plot of the film are likely to show up in a summary of the plot. We have chosen six recent films and sourced their Spanish-language summary from their respective Wikipedia pages. We then ask each topic modelling method to retrieve the top 10, 20, 50, and 100 words and examine how many of the retrieved words are also present in the summary. In other words, our primary evaluation metric is the recall of the retrieval performance. We additionally pay special attention to the 50-word retrieval performance, as some summaries have vocabulary size that barely exceeds 100.

Table 2 shows some basic information about the movies used.

5.2 Hyperparameter Tuning

For both LSA and LDA, the most important hyperparameter that needs to be tuned is the number of topics used in the model. Since LSA uses SVD under the hood, we can simply use the SVD's explained variance as the metric to be optimized. For LDA however, there is no similar built-in metric and so we default to using the retrieval recall as a guidance for selecting the number of topics.

For LSA, we tested a range of topics number ranging from 5 to 50. We attempt to apply the elbow method to this visualization but there is no clear

elbow even with a large amount of topics. For the rest of our analysis, we will use 50 topics for LSA. See figure 2.

For LDA, there is a clear elbow when the number of topics is 3. We will use this hyperparameter for the rest of our evaluation. See figure 3.

5.3 Results

We evaluated the different topic modelling methods for retrieval recall both with (figure 4) and without (figure 5) stop words. LDA clearly provides the best performance with an 80% 50-word retrieval recall, followed by using raw term frequency at about 60% 50-word retrieval recall. LSA and our custom tf-idf implementation both yielded comparable but inferior retrieval performance.

6 Team Member Contributions

Casper:

- Synthesized previous work and designed the sentence ranking (excluding SRS) process.
- Researched and finalized sensible approaches for topic modelling.
- Implemented the end-to-end pipeline that takes in vtt files, computes all per-token metrics, and outputs to JSON format which can then be ingested by the database and the front-end.
- Devised the evaluation method, implemented and benchmarked all topic modelling algorithms and produced all visualizations.
- Designed and produced most parts of the project expo poster.

Nick:

- Initiated the project concept and guided its development, providing the primary vision and objectives for the Spaced Repetition System (SRS) and the language learning application
- Developed the complete backend logic, including the core SRS functionalities, user management, and database interactions
- Responsible for the algorithms for adaptive learning, including token difficulty adjustment and flashcard review scheduling

7 Project Code

The project code is stored in our GitHub repo.

References

[1] N. J. Cepeda, H. Pashler, E. Vul, J. T. Wixted, and D. Rohrer, "Distributed practice in verbal recall tasks: A review and quantitative synthesis.," *Psychological Bulletin*, vol. 132, no. 3, pp. 354–380, 2006, ISSN: 0033-2909. DOI: 10.1037/0033-2909.132.3.354. [Online]. Available: http://dx.doi.org/10.1037/0033-2909.132.3.354.

[2] H. P. Bahrick, L. E. Bahrick, A. S. Bahrick, and P. E. Bahrick, "Maintenance of foreign language vocabulary and the spacing effect," *Psychological Science*, vol. 4, no. 5, pp. 316–321, 1993. DOI: 10.1111/j.1467-9280.1993.tb00571.x. [Online]. Available: https://doi.org/10.1111/j.1467-9280.1993.tb00571.x.

[3] A. Tolmachev, S. Kurohashi, and D. Kawahara, "Automatic japanese example extraction for flashcard-based foreign language learning," *Journal of Information Processing*, vol. 30, pp. 315–330, 2022. DOI: 10.2197/ipsjjip.30.315.

[4] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, N. Calzolari, K. Choukri, T. Declerck, *et al.*, Eds., Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 923–929. [Online]. Available: https://aclanthology.org/L16-1147.

[5] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," To appear, 2017.

[6] *El corpus de referencia del español actual (CREA)*, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, 2014. [Online]. Available: http://hdl.handle.net/11372/LRT-895.

[7] *Vocabulary Studies in First and Second Language Acquisition*. Palgrave Macmillan UK, 2009, ISBN: 9780230242258. DOI: 10.1057/9780230242258. [Online]. Available: http://dx.doi.org/10.1057/9780230242258.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," May 2010, pp. 45–50. DOI: 10.13140/2.1.2393.1847.

Tables

Word Frequency Rank	CEFR Level	Language Importance Score
Top 2500	A1, A2	1
Top 2500 - 3750	B1, B2	0.75
Top 3750 - 5000	C1, C2	0.5
Else	N/A	0.25

Table 1: Correspondence between vocabulary size, CEFR levels, and language importance score

Name	Release	Genre	Movie Vocab Size	Summary Vocab Size
8 Mile	2002	Drama	1887	468
Spider-Man	2023	Superhero	2249	385
The Social Dilemma	2020	Docudrama	2770	117
The Social Network	2010	Biography	2735	443
The Tinder Swindler	2022	Documentary	2345	126
Whiplash	2014	Drama	1329	323

Table 2: Basic information about the films used for evaluation

Figures

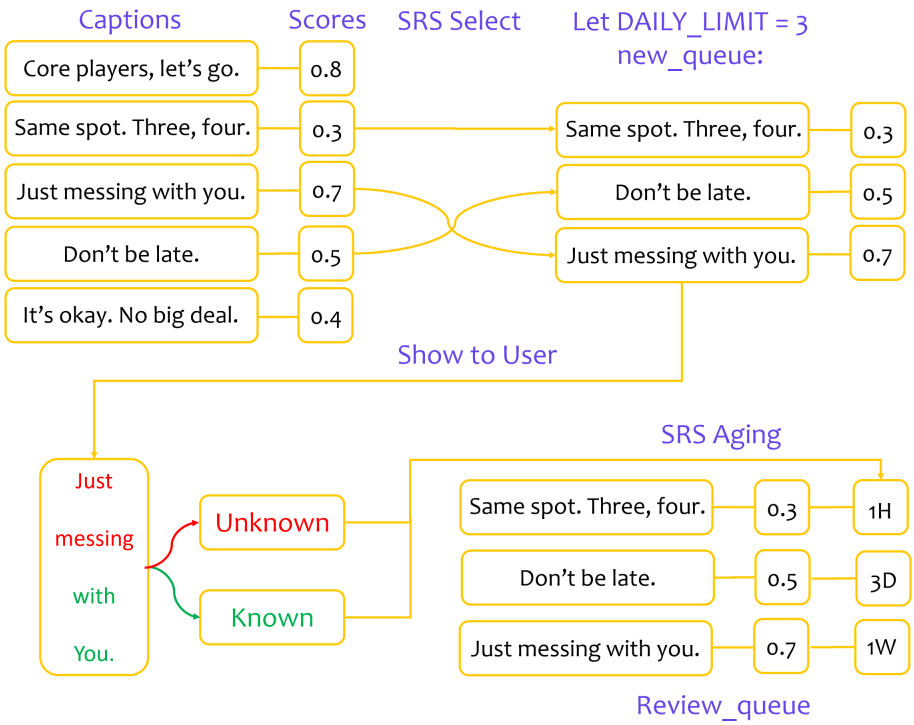

Figure 1: Schematic of *Maestro*'s SRS

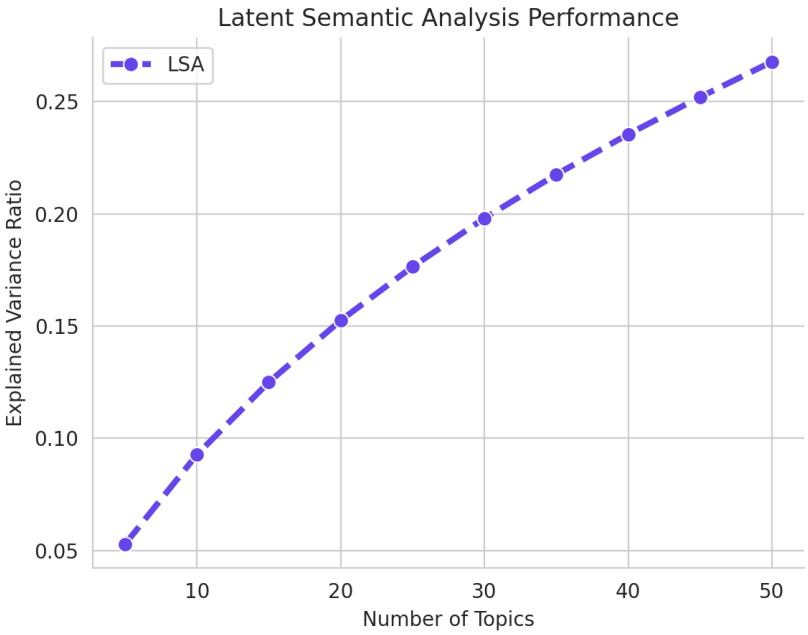

Figure 2: LSA Explained Variance by Number of Topics

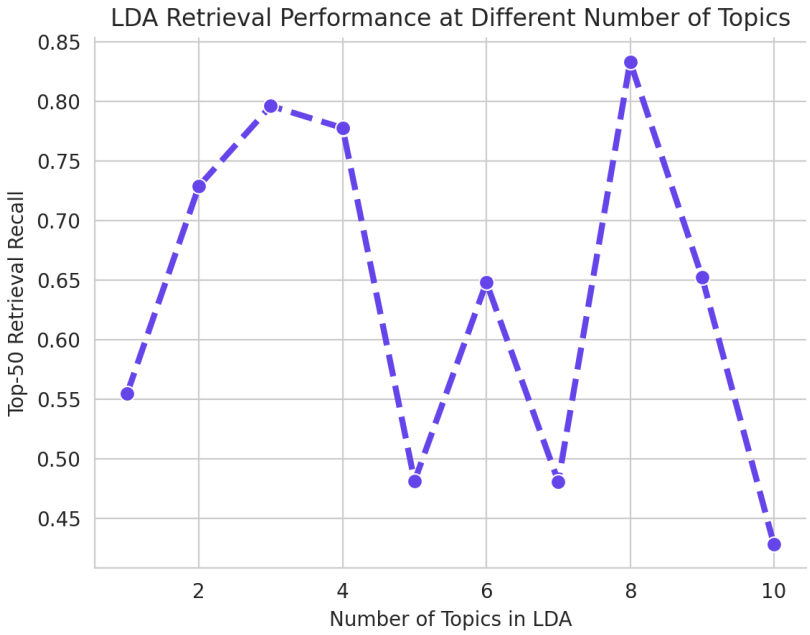

Figure 3: LDA Retrieval Performance by Number of Topics

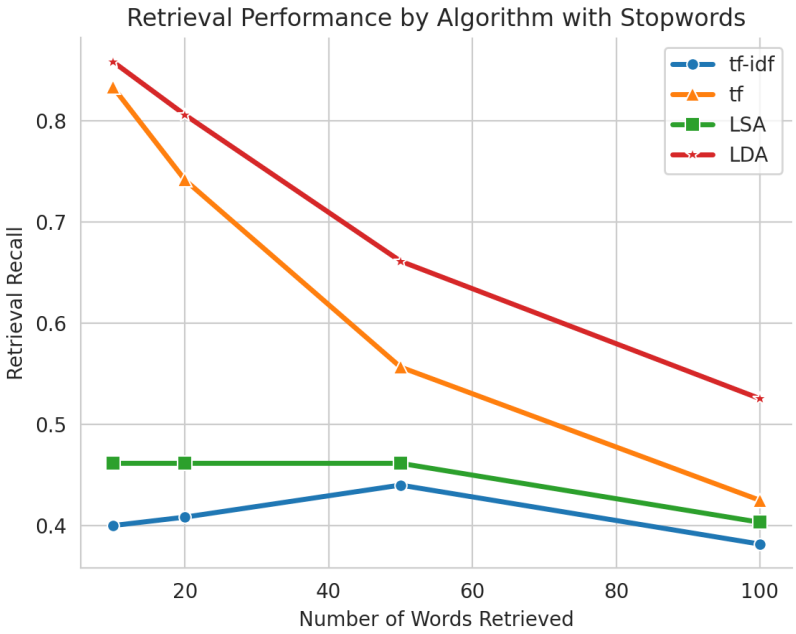

Figure 4: Retrieval Performance by Method Including Stop Words

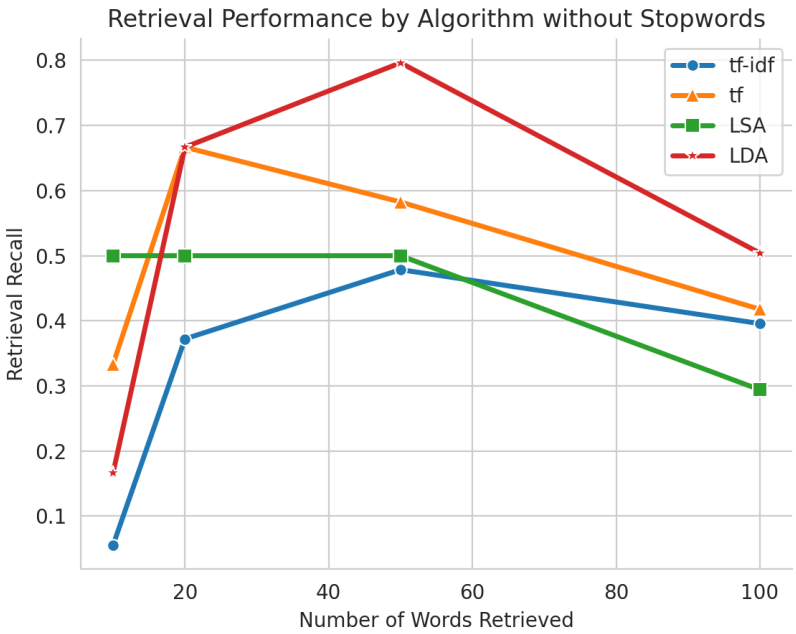

Figure 5: Retrieval Performance by Method Excluding Stop Words

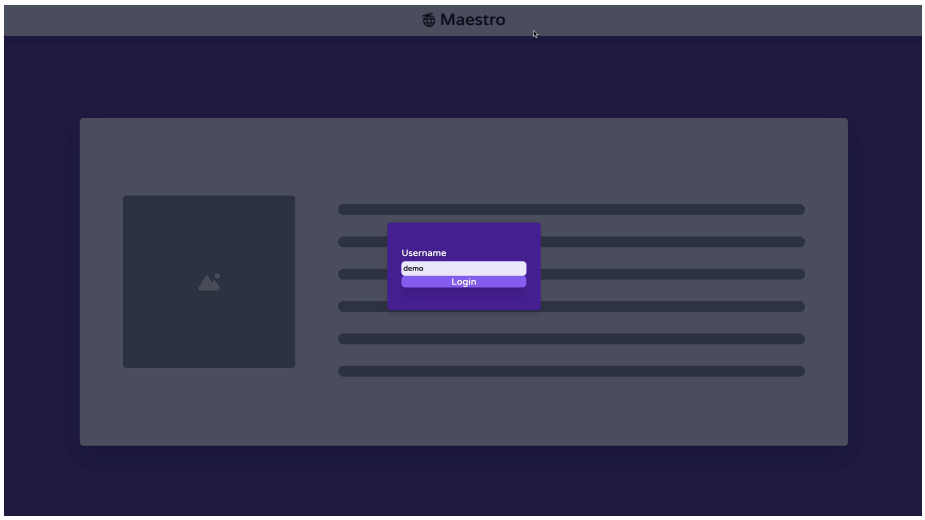

Figure 6: Maestro System Log-in Splash

Figure 7: Selecting Unknown Words Upon Review

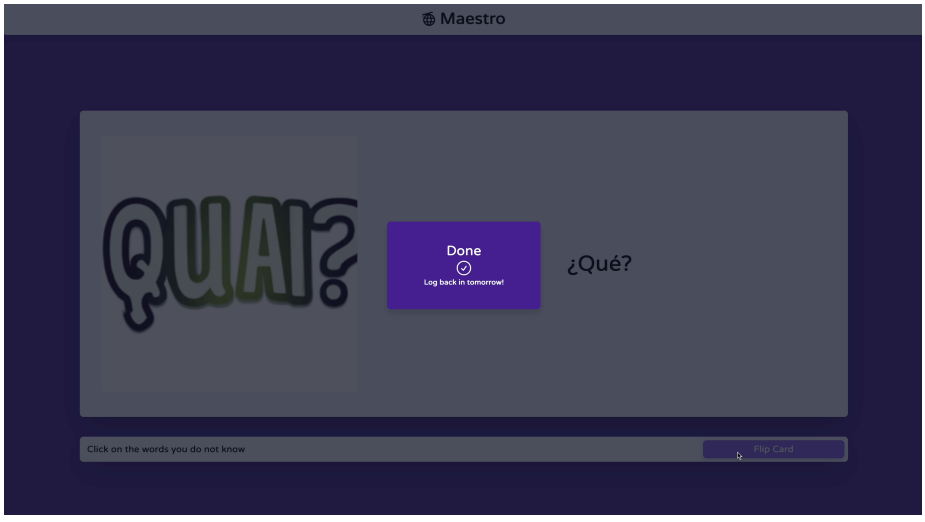

Figure 8: Splash at End of Review Session