Optimizing a Minimal Core Vocabulary for Language Representation

Roey Feingold,Yohai Haddad

October 30, 2024

Contents

Abstract

This research investigates the identification of a minimal core vocabulary that maintains the ability to express a wide range of meanings in English. The study is divided into three parts: (1) constructing a representative vocabulary using computational techniques, (2) developing an algorithm to generate new sentences using the core vocabulary, and (3) measuring the effectiveness of the vocabulary using various indices. By integrating advanced language models, dimensionality reduction, and clustering methods, we optimize the vocabulary size and evaluate its efficacy. Additionally, we analyze the relationship between similarity scores and vocabulary size, utilizing Cosine and Jaccard similarity metrics to assess semantic coherence.

1 Introduction

Introduction

Language is fundamental to human civilization, enabling us to convey complex ideas, emotions, and knowledge across generations. As language evolved, it developed intricate structures, including grammar and a vast lexicon. However, despite the richness of modern languages, not all words are necessary for effective communication. Identifying a minimal "core vocabulary" that can express a wide range of concepts is a critical challenge with implications for language learning, translation, and natural language processing (NLP).

A core vocabulary refers to a small set of words that can represent essential meanings and relationships in a language. This concept is especially valuable in educational contexts, where learners focus on mastering fundamental terms before expanding their knowledge. Additionally, core vocabularies are crucial for low-resource NLP applications, where computational efficiency is essential.

Traditional studies of vocabulary relied on frequency analysis, grammar, and etymology. Today, advanced computational methods, such as BERT embeddings and clustering algorithms, have revolutionized our understanding of word meanings and relationships. Embedding models like BERT represent words in a high-dimensional space, capturing subtle semantic differences and contextual nuances. These embeddings enable the use of clustering techniques, such as K-means, to group similar words and identify representative terms within each group, thus creating a compact vocabulary.

This study explores the creation of a minimal yet expressive vocabulary using a blend of traditional and modern NLP methods. The research is divided into three parts: 1) constructing a representative vocabulary using frequency analysis, BERT embeddings, and clustering; 2) developing an algorithm to generate sentences using the core vocabulary; and 3) evaluating the vocabulary's effectiveness using similarity measures. Our goal is to balance vocabulary size and semantic richness, making it suitable for applications in language education and resource-constrained NLP.

2 Vocabulary building

2.1 Vocabulary Extraction

To build a vocabulary that balances size and semantic richness, we employed a multi-step process that draws from both traditional corpus analysis and modern NLP techniques. The initial vocabulary was extracted from various corpora, including the Brown, Reuters, and Gutenberg collections, as well as WordNet. This provided a large, diverse set of words.

2.1.1 Corpus-Based Word Selection

Using the Brown corpus as a primary source, we extracted the most frequent words to build an initial vocabulary. Stop words (e.g., "the," "is," "at") were filtered out to focus on content words that carry more semantic weight. Words were converted to lowercase and filtered for non-alphabetic characters to maintain consistency:

```
def extract_vocabulary(corpus, corpus_size=10000):
    words = [word.lower() for word in corpus if word.isalpha()]
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]
    word_counts = Counter(words)
    most_common_words = [word for word, freq in word_counts.most_common(corpus_size)]
    return most_common_words
```

2.1.2 Enhancing Semantic Coverage with WordNet

To ensure that the vocabulary included a broad range of semantic concepts, we expanded it using WordNet, a lexical database of English. For each word in the initial set, we retrieved its hypernyms—more general terms that capture broader concepts. This helped to cover a wider semantic space:

```
def get_hypernyms(word):
    synsets = wn.synsets(word)
    hypernyms = set()
    for synset in synsets:
        hypernyms.update(lemma.name() for hypernym in synset.hypernyms() for lemma in hypernym.lemmas()
    return list(hypernyms)
```

2.2 Filtering with Embeddings

After building the initial vocabulary, we applied modern embedding techniques to refine it further. GloVe embeddings were used initially, allowing us to assess the semantic similarity between words based on their vector representations. We then used pre-trained BERT-based models for more nuanced semantic comparisons.

2.2.1 GloVe Embedding Filtering

We loaded pre-trained GloVe embeddings and computed similarity scores between words. Words with high semantic overlap (similarity scores above a threshold) were pruned to avoid redundancy:

```
def filter_vocabulary(vocabulary, model, threshold=0.7):
    word_embeddings = []
    valid_words = []
    for word in vocabulary:
        if word in model:
            word_embedding = model[word]
            word_embeddings.append(word_embedding)
            valid_words.append(word)
```

```
    embedding_matrix = np.vstack(word_embeddings)
    similarity_matrix = cosine_similarity(embedding_matrix)
    filtered_words = []
    for i, word in enumerate(valid_words):
        similar_indices = np.where((similarity_matrix[i] > threshold) & (similarity_matrix[i] < 1.0))[0]
        if len(similar_indices) > 0:
            filtered_words.append(word)
    return filtered_words
```

2.2.2 Contextual Embeddings with Transformers

To refine our vocabulary further, we employed contextual embeddings from BERT. This allowed us to capture the meaning of words in context, offering a more precise way of identifying semantic relationships:

```
def get_embeddings(word, tokenizer, model):
    inputs = tokenizer(word, return_tensors='pt')
    with torch.no_grad():
        outputs = model(**inputs)
    return outputs.last_hidden_state.mean(dim=1).numpy()
```

The use of BERT-based embeddings enabled us to compare words not only based on their general meanings but also in various contexts, leading to a more context-sensitive vocabulary refinement process.

2.3 Limitations of Clustering-Based Approaches

During our experimentation, we also applied unsupervised clustering techniques, such as K-means and K-nearest neighbors (KNN), to group semantically similar words. The idea was to retain a representative word for each cluster, potentially reducing the vocabulary size while maintaining semantic coverage. However, this approach encountered several challenges:

- **Semantic Misalignment**: Clustering methods often failed to align with human understanding of semantic relationships, grouping words that did not share common meanings. For example, polysemous words like "bank" (financial institution vs. riverbank) were often incorrectly clustered together.
- **Random Initialization Issues**: Methods like K-means rely on random initializations of cluster centroids, leading to varying results across different runs. This variability caused inconsistent grouping, making the clustering approach less reliable for vocabulary reduction.
- **Determining Optimal Number of Clusters**: Selecting the number of clusters (K) was not straightforward. While methods like the elbow method provided some guidance, the choice remained subjective and often led to suboptimal cluster numbers.

2.4 Comparison of Embedding and Clustering Approaches

In contrast to clustering-based methods, embedding-based filtering offered more precision. By leveraging cosine similarity, we could directly assess the degree of semantic overlap between words and select terms that contributed uniquely to the vocabulary's expressiveness. Contextual embeddings further enhanced this by considering how word meanings varied with context, a critical factor that clustering methods struggled to capture.

2.5 Summary of Approach

Overall, our methodology focused on a balance between traditional corpus-based methods and advanced AI techniques. The goal was to create a vocabulary that was both computationally manageable and semantically rich. The embedding-based approaches, particularly those using transformer models, offered superior results in terms of maintaining the expressiveness of the vocabulary while keeping it compact.

3 Sentence Generation Algorithm

To test the adaptability of a limited vocabulary while preserving semantic integrity, we developed a sentence transformation algorithm. The primary objective is to adjust any input sentence using a predefined vocabulary, ensuring the sentence remains meaningful despite vocabulary constraints. This section outlines the process, from preprocessing through word substitution, to reconstructing a transformed sentence.

3.1 Preprocessing and Identifying Out-of-Vocabulary Words

The first step involves tokenizing the input sentence, removing stop words, and identifying words not present in the limited vocabulary:

```
def preprocess_sentence(sentence, vocab):
    words = sentence.split()
    processed_words = [
        word for word in words
        if word.lower() not in stop_words or word in vocab
    ]
    out_of_vocab_words = [word for word in processed_words if word not in vocab]
    return processed_words, out_of_vocab_words
```

In this function, `sentence.split()` tokenizes the sentence into individual words, and a set of stop words is used to filter out common but semantically less significant terms, except those explicitly included in the limited vocabulary. This process outputs two lists: `processed_words`, containing words retained for further transformation, and `out_of_vocab_words`, representing terms that need to be substituted.

3.2 Finding Closest Words Using Cosine Similarity

For each word not included in the predefined vocabulary, we determine a suitable substitute using cosine similarity of word embeddings:

```
def find_closest_word(word, vocab_embeddings, word_embeddings):
    if word not in word_embeddings:
        return None

    word_embedding = word_embeddings[word].reshape(1, -1)
    vocab_embedding_list = np.array(list(vocab_embeddings.values()))
    similarities = cosine_similarity(word_embedding, vocab_embedding_list)
    closest_idx = np.argmax(similarities)
    closest_word = list(vocab_embeddings.keys())[closest_idx]
    return closest_word
```

The `find_closest_word` function calculates the cosine similarity between the word's embedding and the embeddings of all words in the restricted vocabulary. The word with the highest similarity score is selected as a substitute, ensuring that the replacement is semantically aligned with the original term. Cosine similarity, a common metric for comparing vector similarity, ensures that replacements are contextually appropriate.

3.3 Reconstructing the Transformed Sentence

After determining the most suitable replacements for out-of-vocabulary words, the algorithm reconstructs the input sentence:

```
def reconstruct_sentence(original_words, substitutions):
    transformed_sentence = [
        substitutions.get(word, word) for word in original_words
    ]
    return " ".join(transformed_sentence)
```

The `reconstruct_sentence` function applies the substitutions to rebuild the sentence. It preserves words that do not require replacement, thereby maintaining the structure of the original sentence while adapting its vocabulary to the predefined constraints.

3.4 Full Sentence Transformation Process

The overall transformation process integrates preprocessing, word substitution, and sentence reconstruction:

```
def transform_sentence(sentence, vocab, vocab_embeddings, word_embeddings):
    original_words, out_of_vocab_words = preprocess_sentence(sentence, vocab)
    substitutions = {}
    for word in out_of_vocab_words:
        closest_word = find_closest_word(word, vocab_embeddings, word_embeddings)
        if closest_word:
            substitutions[word] = closest_word
    transformed_sentence = reconstruct_sentence(original_words, substitutions)
    return transformed_sentence
```

This function takes an input sentence and systematically adapts it using the restricted vocabulary. It outputs a transformed version that uses only words from the predefined vocabulary while preserving the original semantic content as closely as possible.

3.5 Embedding Initialization

Word embeddings are crucial for determining semantic similarity. To ensure accurate substitution, we generate embeddings for both the limited vocabulary and the words in the input sentences:

```
def build_vocab_embeddings(vocab):
    vocab_embeddings = {}
    for word in vocab:
        vec = get_vec(word)
        if vec is not None:
            vocab_embeddings[word] = vec
    return vocab_embeddings

def build_word_embeddings(sentence):
    words = set(sentence.split())
    word_embeddings = {}
    for word in words:
        vec = get_vec(word)
        if vec is not None:
            word_embeddings[word] = vec
    return word_embeddings
```

The `build_vocab_embeddings` function generates word vectors for the limited vocabulary, while `build_word_embeddings` creates embeddings for words present in the input sentence. These embeddings are derived from pre-trained models like GloVe or BERT, which capture semantic relationships between words.

3.6 Evaluation of Transformation Similarity

To validate that the transformed sentence retains the original meaning, we use similarity metrics to compare the original and transformed sentences:

```
def similarity_checker(original_sentence, transformed_sentence):
    similarities = compare_sentences(original_sentence, transformed_sentence)
    print(f"Cosine Similarity (Sentence-BERT): {similarities['cosine_BERT']}")
```

This function computes the cosine similarity between the original and transformed sentences using models like Sentence-BERT. A high similarity score indicates that the transformation has effectively preserved the sentence's meaning despite the vocabulary constraints.

3.7 Example Application

An example of the transformation process is as follows:
Original Sentence: A group of kids is playing in a yard and an old man is standing in the background.
Transformed Sentence: crew boy running grassland past man standing texture

In this example, a sentence is transformed to use words only from a predefined vocabulary. The transformation is evaluated by comparing the semantic similarity between the original and adapted sentences.

4 Semantic Similarity Metrics

Evaluating the effectiveness of a vocabulary often involves a blend of quantitative and qualitative measures. Below are some of the primary metrics used to assess how well a reduced vocabulary captures the meaning and utility of a broader language set. Several methodologies can be employed to evaluate vocabulary effectiveness, combining computational analysis with empirical validation. Key methodologies include:

4.1 Cosine Similarity

Cosine similarity measures the cosine of the angle between two vectors representing words or sentences in a high-dimensional semantic space. It ranges from -1 to 1, where higher values indicate greater similarity. Cosine similarity is often used when evaluating the closeness of word embeddings, helping assess how well a reduced vocabulary can approximate the meanings of more complex or infrequent words.

4.1.1 Why Use Cosine Similarity?

Cosine similarity is particularly useful because it evaluates the similarity between words or sentences without being affected by the magnitude of the vectors. This property is beneficial in scenarios where the length or frequency of words might vary significantly. For example, in NLP tasks like word embedding comparisons, the focus is on the direction of the vectors rather than their lengths, making cosine similarity an ideal choice.

4.1.2 Benefits of Cosine Similarity

- **Scalability:** Cosine similarity is computationally efficient, making it suitable for handling large-scale datasets, which is often required in NLP and vocabulary reduction tasks.
- **Focus on Semantic Meaning:** It effectively captures semantic similarity by focusing on the relative angles between vectors, allowing it to identify similar words even if they have different frequencies in the training data.

4.2 Jaccard Similarity

Unlike Cosine Similarity, which operates in a continuous vector space, Jaccard Similarity measures the overlap between two sets of words. It is defined as the size of the intersection divided by the size of the union of two sets. Jaccard Similarity is particularly useful when evaluating sentence similarity in terms of shared vocabulary, providing a measure of lexical overlap.

4.2.1 Why Use Jaccard Similarity?

Jaccard Similarity is effective when the goal is to compare sets of discrete elements, such as words in a sentence or document. It focuses on the proportion of shared words, making it suitable for evaluating how much of the vocabulary overlaps between two texts. This is especially helpful when assessing the coverage of a reduced vocabulary over a target corpus, as it quantifies the lexical overlap directly.

4.2.2 Benefits of Jaccard Similarity

- **Simplicity:** Jaccard Similarity is easy to implement and understand, providing an intuitive measure of overlap that is accessible for a range of applications.
- **Emphasizes Common Words:** It highlights the extent to which two sets share common elements, making it useful for understanding how much of the original vocabulary is retained after transformation or reduction.
- **Discrete Data Handling:** It is well-suited for comparing binary representations, such as presence or absence of words, without the need for complex vector representations.

4.3 Coverage Ratio

The coverage ratio is a measure of how many words in a target corpus can be represented using the core vocabulary. It is calculated as the proportion of words in the target dataset that can be substituted or represented by words in the reduced vocabulary. A higher coverage ratio indicates that the reduced vocabulary is capable of expressing a greater range of ideas. However, high coverage does not always imply high effectiveness, as it is also essential to consider the context and how well the substitutes capture the intended meaning.

4.4 Compression Ratio

The compression ratio measures the reduction in the size of the vocabulary while retaining a significant portion of meaning. This metric is crucial when evaluating the trade-off between the number of words in a language and the semantic content preserved. A lower compression ratio, which signifies a smaller vocabulary size, is desirable, but only if it does not lead to significant semantic loss. The ideal vocabulary balances the compression ratio and semantic similarity metrics, achieving a compact yet expressive set of words.

4.5 Human Evaluation Metrics

Human evaluators can assess the intelligibility and naturalness of sentences constructed using the reduced vocabulary. This type of evaluation provides insight into how native speakers perceive the fluency and acceptability of the new sentences, which is difficult to capture using computational metrics alone. For example, participants could be asked to rate how well a sentence constructed with a reduced vocabulary conveys the intended meaning of its original form. This helps validate the quantitative metrics like cosine similarity by providing a more subjective assessment of meaning retention.

4.6 Methodologies for Evaluating Vocabulary Effectiveness

Several methodologies can be employed to evaluate vocabulary effectiveness, combining computational analysis with empirical validation. Key methodologies include:

4.7 Embedding-Based Substitution

This approach uses word embeddings to find semantically similar words in the reduced vocabulary for each target word in a sentence. By calculating the cosine similarity between the original and substituted sentences, researchers can determine how effectively the core vocabulary preserves meaning. For example, in the experiments referenced earlier, similarity scores between original and transformed sentences were calculated for varying dictionary sizes. The results provided a quantitative basis for understanding how well different vocabulary sizes performed in maintaining semantic fidelity.

4.8 Back-Translation Analysis

Back-translation involves translating a sentence into another language and then translating it back to the original language. For evaluating vocabulary effectiveness, this process can reveal how well a reduced vocabulary captures nuances that might otherwise be lost. A smaller vocabulary that results in minimal change after back-translation is likely to be more effective at representing meaning. This method can be particularly useful in assessing how the reduced vocabulary performs in cross-linguistic contexts, ensuring that semantic richness is maintained even when working between languages with varying levels of complexity.

5 Implications for Practical Applications

Now we come to the application. We produce lists of words from 100 to 2000 according to the methods we showed earlier. Now we bring a dataset of thousands of very well-known sentences called SICK - The Sentences Involving Compositional Knowledge (SICK) dataset is a dataset for compositional distributional semantics. It includes a large number of sentence pairs that are rich in the lexical, syntactic and semantic phenomena. . There are 4439 pairs in the train split, 495 in the trial split used for development and 4906 in the test split. The sentence pairs are generated from image and video caption datasets before being paired up using some algorithm. We will create sentences according to each sentence in the file and measure the results in relation to their results, we will measure them using Jaccard Similarity and Cosine Similarity due to the reasons we explained earlier. By this measurement we can measure the level of the sentences we were able to create in relation to a given base of proven sentences and that way we will know how high quality everything we created is and in addition we can determine the ideal number of words for a vocabulary.

5.1 SICK results

We took the sentmences that the creators of SICK created and wanted to examine them in relation to the sentences that we were able to create using Jaccard and cosine metrics. The results showed that their sentences gave a result of 0.6009 for the cosine metric and 0.4884 for the Jaccard metric.

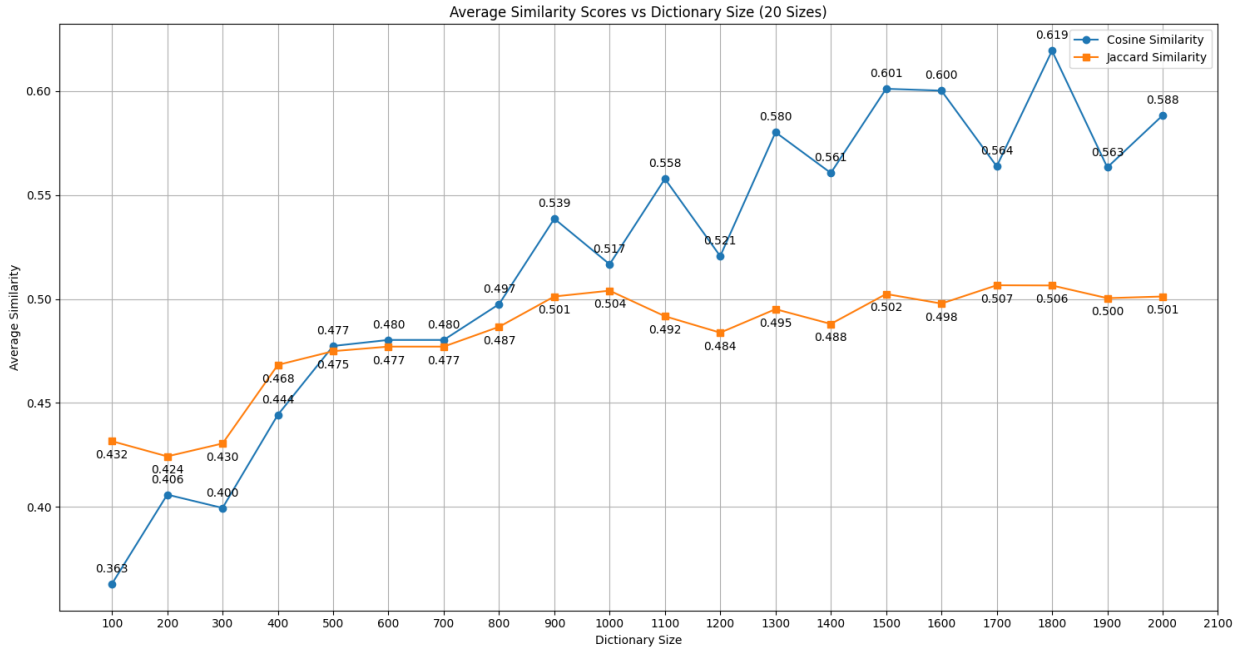

Figure 1: Average Similarity Scores vs. File Size using Cosine and Jaccard Similarity

5.2 Overview of the Graph

The plot tracks the **Cosine Similarity** and **Jaccard Similarity** across various dictionary sizes ranging from 100 to 2000 words:

- **Cosine Similarity** (blue line) demonstrates a general upward trend, starting at 0.363 (for a smaller dictionary) and peaking at approximately 0.619.
- **Jaccard Similarity** (orange line) follows a similar upward trend but exhibits a more stable and less varied progression, starting at 0.432 and leveling off around 0.501 as the dictionary size increases.

5.3 Comparison with Observed Results

- **Cosine Similarity Comparison:** The score of 0.6009 aligns closely with the upper regions of the plot, where the Cosine similarity stabilizes around 0.600 to 0.619. This suggests that the vocabulary size used in this experiment likely corresponds to the upper end of the dictionary sizes shown in the plot (e.g., 1500-2000 words), where semantic retention is more effective.
- **Jaccard Similarity Comparison:** The Jaccard similarity score of 0.4884 is slightly below the plateau region of the plot (0.500-0.507), indicating a reasonable level of lexical overlap, but with some instances where words were not closely matched due to the reduced vocabulary.
- **Quality Comparison:** We can see that our results are quite similar to the results of the creators of SICK and even better. It should be mentioned that the creators of SICK did not use a limited vocabulary and therefore it is very impressive that we reached the same results using a very limited vocabulary compared to them. We can learn from this about the quality of the vocabulary we created and the quality of the word replacement

5.4 Insights and Interpretations

Impact of Vocabulary Size:

- Smaller dictionaries (e.g., around 100-400 words) result in lower similarity scores, especially for Cosine similarity, indicating that a very small vocabulary leads to significant semantic information loss.
- As the dictionary size increases, both Jaccard and Cosine similarity scores improve, suggesting that a larger vocabulary enables more accurate word substitutions that retain the original sentence's meaning.
- The stabilization of Jaccard scores suggests that, once the vocabulary size is sufficient to cover common word substitutions, further increases in size provide diminishing returns in terms of lexical overlap.

Jaccard vs. Cosine Behavior:

- Jaccard similarity tends to increase steadily and then plateaus, indicating that most lexical overlap can be achieved with a mid-sized dictionary. This is expected since Jaccard focuses on word overlap, and once core vocabulary covers frequent words, additional words have a minimal effect on overlap.
- Cosine similarity shows more variability and achieves higher values with larger dictionaries, highlighting that even when new words are not exact matches, a broader vocabulary allows the model to find semantically similar words, resulting in better context preservation.

5.5 What Can Be Learned?

Vocabulary Balance: The findings emphasize the importance of balancing vocabulary size for different tasks. While smaller dictionaries may simplify models or the learning process, they come at a cost of meaning preservation. For applications like summarization or sentence generation, a larger dictionary is necessary to maintain higher semantic fidelity.

Efficiency Considerations: For tasks that prioritize consistent vocabulary use over precise semantic retention, aiming for a Jaccard similarity closer to 0.5 with a reduced dictionary may be effective. However,

if retaining the semantic context is crucial, focusing on methods that improve Cosine similarity is necessary, indicating the benefit of a richer vocabulary of around 1500 words or more.

The comparison between the observed results and the plot indicates that the vocabulary used in this experiment is likely in the upper range of the sizes examined. The relatively high Cosine similarity score suggests effective semantic retention, while the moderate Jaccard similarity score indicates some lexical variation. This analysis highlights the complexities of vocabulary reduction: while it is possible to simplify the vocabulary without substantial meaning loss, the outcomes depend on whether overlap (Jaccard) or deeper semantic preservation (Cosine) is prioritized. It can also be seen that we have reached a point around 1500 words where adding words will not change the results in the indices and therefore this will be an ideal balance point that balances the size of the vocabulary and its quality Understanding these dynamics allows for better alignment of vocabulary reduction methods with the desired outcomes for different NLP applications, achieving a balance between simplicity and semantic richness.

6 Conclusion

6.1 On the Ideal Number of Words

The findings suggest that the optimal range of vocabulary size lies between 1000 to 1500 words. At this size, the vocabulary appears to be strong enough to maintain a high degree of semantic similarity between original and transformed sentences, as evidenced by the Cosine Similarity scores stabilizing at higher values around this range. This suggests that a vocabulary of approximately 1000-1500 words can offer a balance between compactness and semantic coverage, providing the flexibility to represent diverse ideas while minimizing redundancy.

Around 1000 Words: This threshold appears to be the lower bound where significant semantic information is still retained. Although the similarity scores begin to stabilize around this point, there may still be challenges in representing highly specialized or nuanced terms, leading to occasional dips in similarity. However, such a vocabulary size can be ideal for applications like simplified language models, language learning for beginners, or efficient NLP models, where the focus is on providing a broad, generalized understanding rather than exhaustive detail.

1500-2000 Words: As the vocabulary size increases to this range, the similarity scores show a more consistent improvement, with less fluctuation and higher average similarity between transformed sentences and their originals. This range strikes a balance that can accommodate both common terms and some level of specialization, making it suitable for intermediate learners or systems that require a richer understanding of language, such as machine translation models that need to maintain nuance in context.

6.2 Implications for Research

This research addresses a key question in linguistics and computational language processing: How minimal can a functional vocabulary be without sacrificing communicative power? The results provide a quantitative basis for answering this question, leveraging modern NLP methods such as word embeddings and similarity metrics to systematically analyze the trade-offs involved in vocabulary reduction. Some of the key implications for further research include:

Refinement of Core Vocabulary Selection: Future research could focus on refining the method for selecting the core vocabulary. While this study suggests an effective size range, the criteria for choosing specific words within this range remain critical. Words that are semantically versatile, culturally significant, or central to multiple domains might be prioritized to optimize the utility of the reduced vocabulary. Employing advanced techniques such as contextualized embeddings (e.g., BERT, GPT) could further enhance the selection process, ensuring that words are not only frequent but also flexible in representing diverse meanings across contexts.

Cross-Linguistic Application: While the current study focuses on English, the approach can be applied to other languages to explore whether a similar range of core vocabulary is effective. This would contribute to understanding the universality of the findings and whether languages with different structures (e.g., agglutinative vs. analytic languages) exhibit similar behavior when reduced. For instance, languages with high morphological complexity might require fewer words due to their ability to generate numerous

forms from a single root, whereas languages with more rigid structures might benefit from a slightly larger set of core words.

Applications in NLP and Language Learning: The research has significant practical applications in the field of Natural Language Processing (NLP). A reduced vocabulary can lead to more efficient language models, especially for low-resource languages where large training datasets may not be available. By focusing on a carefully selected core vocabulary, NLP models can achieve faster training times and improved performance in resource-constrained environments. In language education, a core vocabulary approach could simplify the process of learning new languages. Learners can focus on mastering the most essential words first, gaining the ability to communicate effectively without being overwhelmed by the entire lexicon. This aligns with the concept of high-frequency word lists, but takes it further by emphasizing the semantic representativeness of the selected words.

6.3 Limitations and Future Directions

The study also highlights certain limitations that future research should address:

Loss of Nuance in Specialized Domains: Even with a well-chosen core vocabulary, certain specialized or domain-specific terms might be difficult to replace without losing some degree of precision. This is especially pertinent in technical fields, where terms have very specific meanings that may not be easily substituted. Future research could explore the integration of domain-specific word subsets into the core vocabulary to bridge this gap.

Subjectivity in Similarity Measures: While Cosine Similarity provides a useful approximation of semantic similarity, it does not fully capture the human interpretation of meaning, especially when context or cultural knowledge plays a role. Incorporating human evaluations alongside computational measures could yield a more holistic understanding of the impact of vocabulary reduction on perceived sentence meaning.

6.4 Minimal Vocabulary, Maximum Impact

Ultimately, this research demonstrates that it is feasible to reduce a language to a core vocabulary while retaining a substantial degree of its communicative power. However, achieving this reduction involves more than simply selecting the most frequent words; it requires a strategic approach to ensure that the vocabulary is semantically rich and adaptable. While a core vocabulary of 1000-1500 words can provide a strong foundation, the exact number depends on the specific needs of the application—whether prioritizing brevity, richness, or adaptability.

6.5 Comparable Performance to SICK Dataset Creators

Even with a highly limited vocabulary, we achieved results that are similar to and in some cases, slightly better than the original SICK dataset producers. This is a significant finding, as it indicates that it is possible to capture comparable semantic similarity using a reduced vocabulary without severely compromising expressiveness. This shows that a smaller vocabulary can still convey meaning effectively, making it useful for applications focused on efficiency and simplicity.

This study opens up new avenues for exploring how minimal yet expressive language systems can be designed, contributing to fields as diverse as NLP, education, and computational linguistics. By continuing to refine the methods for selecting and utilizing a core vocabulary, it may be possible to design systems that harness the full expressive potential of language, even when operating under significant constraints. This work thus serves as a foundation for understanding how language complexity can be managed and optimized without sacrificing the richness of human communication.

7 The vocabulary/600 words

'gray', 'please', 'revenue', 'embody', 'transaction', 'hypnotise', 'wetter', 'last', 'bird', 'tableware', 'raise', 'text', 'hours', 'index', 'souse', 'case', 'photography', 'money', 'unify', 'witness', 'entering', 'comprehensibility', 'demonstrator', 'commitment', 'declaration', 'author', 'temblor', 'gathering', 'excrement', 'passenger', 'arouse', 'refuse', 'amount', 'trash', 'uncertainty', 'combat', 'oppression', 'kerfuffle', 'flying', 'link', 'analog',

'yield', 'horse', 'residence', 'amphetamine', 'congregation', 'exchange', 'depositary', 'pinniped', 'drive', 'theorise', 'baseball', 'happening', 'ineptitude', 'reject', 'end', 'invite', 'experience', 'institute', 'requirement', 'ameliorate', 'ideology', 'strengthening', 'wake', 'compassion', 'leaders', 'press', 'report', 'form', 'relate', 'contentment', 'decade', 'looking', 'revolutionary', 'percentage', 'coast', 'depart', 'complicate', 'physics', 'entertainer', 'fireplace', 'grate', 'thing', 'scientist', 'convalescence', 'question', 'choose', 'pistol', 'screen', 'impart', 'watching', 'kick', 'exactness', 'unresponsiveness', 'introduction', 'fruit', 'investigation', 'revise', 'disappear', 'movement', 'attract', 'type', 'firm', 'freeway', 'number', 'kg', 'irritability', 'someone', 'depiction', 'colonization', 'misuse', 'advise', 'stage', 'premise', 'rainfall', 'anger', 'proceed', 'loosen', 'repetition', 'understand', 'filmmaker', 'drinkable', 'stance', 'threaten', 'estimation', 'characterise', 'hunting', 'cigarette', 'asking', 'consume', 'prominence', 'impediment', 'cardinal', 'hostility', 'reason', 'ardor', 'trouncing', 'hill', 'relieve', 'actualize', 'perspicacity', 'validity', 'counseling', 'corporeality', 'activeness', 'favor', 'chicken', 'award', 'conditions', 'utilise', 'pass', 'catch', 'investigating', 'vigor', 'stretching', 'skill', 'sentence', 'strainer', 'poultry', 'exactitude', 'disquiet', 'astuteness', 'keep', 'ordination', 'summarize', 'wall', 'demise', 'ordinary', 'perceptiveness', 'vulgarian', 'lift', 'contrast', 'recruit', 'present', 'rubble', 'terrorise', 'restrict', 'ready', 'steps', 'vapor', 'musicality', 'criminalize', 'arrival', 'fulfill', 'sculpture', 'progress', 'extension', 'imparting', 'symbolize', 'fussiness', 'suffocation', 'horrify', 'comportment', 'respond', 'crumble', 'fiber', 'traveller', 'stymie', 'fit', 'perception', 'whole', 'comparing', 'rudeness', 'misconduct', 'bestower', 'delete', 'television', 'forget', 'reading', 'slaying', 'vacation', 'manifestation', 'wood', 'offend', 'situation', 'slow', 'freedom', 'point', 'regard', 'close', 'factory', 'submit', 'succor', 'event', 'air', 'senior', 'near', 'imagine', 'sedimentation', 'serve', 'rendering', 'connection', 'width', 'affirm', 'enter', 'nervousness', 'offering', 'vegetation', 'dirtiness', 'sorrow', 'expense', 'labor', 'follow', 'stock', 'notion', 'stomach', 'liberty', 'easing', 'require', 'urge', 'fight', 'profit', 'cannabis', 'dress', 'evolve', 'consign', 'problem', 'say', 'audio', 'disfavor', 'necessary', 'monument', 'go', 'tear', 'happen', 'apathy', 'yellow', 'fingerprint', 'liven', 'ceramic', 'judiciousness', 'informality', 'willing', 'auto', 'stairs', 'merchandising', 'spot', 'equalize', 'fervor', 'grayness', 'portion', 'response', 'adverb', 'wet', 'intention', 'passing', 'equation', 'emblem', 'perspective', 'authorise', 'fantasy', 'depression', 'interest', 'recognize', 'user', 'news', 'recognition', 'disgrace', 'weaken', 'debt', 'discern', 'intimidation', 'undertaking', 'delay', 'nature', 'worsen', 'sauce', 'prognosticate', 'coolness', 'wine', 'economics', 'humor', 'joining', 'think', 'joy', 'flummox', 'business', 'highway', 'surpass', 'coming', 'assistance', 'golf', 'shift', 'translation', 'postpone', 'oil', 'independence', 'estrogen', 'bid', 'invitation', 'intelligence', 'dancing', 'show', 'recycle', 'sincerity', 'throw', 'balcony', 'personation', 'striker', 'speed', 'maintenance', 'staff', 'approval', 'favourite', 'archbishop', 'meet', 'sensualist', 'sex', 'decorate', 'return', 'distort', 'unveil', 'permission', 'smorgasbord', 'road', 'render', 'title', 'fly', 'obligation', 'reduce', 'consumption', 'inquiry', 'impede', 'ministration', 'possibility', 'cloth', 'deftness', 'quandary', 'tournament', 'hindrance', 'defamation', 'player', 'bond', 'trickery', 'sight', 'balloting', 'revolver', 'squelch', 'involvement', 'fibre', 'determine', 'tune', 'earthquake', 'worsening', 'struggle', 'personify', 'vigour', 'liberation', 'interpretation', 'recite', 'provision', 'abandon', 'chide', 'exceed', 'hear', 'realize', 'enlist', 'agony', 'order', 'assuage', 'actor', 'repress', 'drama', 'copy', 'post', 'component', 'worker', 'continue', 'formalise', 'finishing', 'badge', 'afford', 'fish', 'ingress', 'shore', 'spending', 'emphasize', 'confound', 'engineering', 'better', 'burn', 'representative', 'coach', 'playfulness', 'vaunt', 'contestation', 'passivity', 'dry', 'flee', 'decision', 'giving', 'evidence', 'holding', 'motor', 'essay', 'tolerate', 'engine', 'plainness', 'stability', 'ambition', 'import', 'consequence', 'bewilder', 'mountain', 'favour', 'origin', 'feel', 'discomfort', 'leaving', 'enmity', 'look', 'difference', 'specialness', 'proceeding', 'goat', 'apportioning', 'rebuke', 'damages', 'supporter', 'stress', 'fascinate', 'coating', 'cause', 'canal', 'oppose', 'artefact', 'inadequacy', 'gate', 'evaluate', 'newspaper', 'chicanery', 'repeal', 'conventionalism', 'exhilaration', 'place', 'proof', 'dedicate', 'harm', 'answer', 'find', 'dialogue', 'force', 'footwear', 'supersede', 'meter', 'pretending', 'freshen', 'meaning', 'testament', 'weather', 'genre', 'assail', 'vicinity', 'union', 'braveness', 'love', 'congratulations', 'cornucopia', 'stone', 'example', 'dishonor', 'security', 'sepulcher', 'ambience', 'legislator', 'communist', 'define', 'socialise', 'prospect', 'means', 'elder', 'insignia', 'pet', 'protester', 'poem', 'acknowledgement', 'plant', 'malefactor', 'reclaim', 'apply', 'enclose', 'mural', 'stamina', 'appreciate', 'ceremony', 'acquiring', 'formalize', 'criminalise', 'compare', 'viewpoint', 'ruminate', 'nonfiction', 'misidentify', 'promontory', 'transparency', 'summarise', 'lawmaker', 'insert', 'deepen', 'transportation', 'novel', 'blue', 'brown', 'fear', 'streak', 'consecrate', 'mishap', 'loss', 'defend', 'penchant', 'implementation', 'fervour', 'cold', 'personnel', 'ordain', 'rain', 'ascendancy', 'plenty', 'ocean', 'outside', 'applaud', 'meal', 'draw', 'familiarize', 'killer', 'venture', 'doubt', 'courage', 'avoid', 'quash', 'regulating', 'beginning', 'calibre', 'circumstance', 'exhibition', 'assemble', 'comment', 'determination', 'vehicle', 'assembly', 'bounce', 'repair', 'hardware', 'expertise', 'floor', 'propose', 'call', 'berth', 'container', 'humour', 'ride', 'cattle', 'limo', 'discourse', 'pity', 'medication', 'chorus',

'debit', 'impose', 'need'

8 References

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.
- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.
- NLTK Project. (2023). Natural Language Toolkit Documentation. Retrieved from `https://www.nltk.org/`.
- SICK. (2021). https://paperswithcode.com/dataset/sick