Dynamic Word Recommendation to Obtain Diverse Crowdsourced Paraphrases of User Utterances

Mohammad-Ali Yaghoub-Zadeh-Fard
m.yaghoubzadehfard@unsw.edu.au
University of New South Wales
Sydney, NSW, Australia

Boualem Benatallah
b.benatallah@cse.unsw.edu.au
University of New South Wales
Sydney, NSW, Australia

Fabio Casati
fabio.casati@servicenow.com
Servicenow
Santa Clara, California, USA

Moshe Chai Barukh
mosheb@cse.unsw.edu.au
University of New South Wales
Sydney, NSW, Australia

Shayan Zamanirad
shayanz@cse.unsw.edu.au
University of New South Wales
Sydney, NSW, Australia

ABSTRACT

Building task-oriented bots requires mapping a user utterance to an intent with its associated entities to serve the request. Doing so is not easy since it requires large quantities of high-quality and diverse training data to learn how to map all possible variations of utterances with the same intent. Crowdsourcing may be an effective, inexpensive, and scalable technique for collecting such large datasets. However, the diversity of the results suffers from *the priming effect* (i.e. workers are more likely to use the words in the sentence we are asking to paraphrase). In this paper, we leverage priming as an opportunity rather than a threat: we dynamically generate word suggestions to motivate crowd workers towards producing diverse utterances. The key challenge is to make suggestions that can improve diversity without resulting in semantically invalid paraphrases. To achieve this, we propose a probabilistic model that generates continuously improved versions of word suggestions that balance diversity and semantic relevance. Our experiments show that the proposed approach improves the diversity of crowdsourced paraphrases.

CCS CONCEPTS

• **Human-centered computing → User interface toolkits**; • **Information systems → Crowdsourcing**;

KEYWORDS

Crowdsourcing, bots, paraphrasing

ACM Reference Format:
Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Fabio Casati, Moshe Chai Barukh, and Shayan Zamanirad. 2020. Dynamic Word Recommendation to Obtain Diverse Crowdsourced Paraphrases of User Utterances. In *25th International Conference on Intelligent User Interfaces (IUI '20), March 17–20, 2020, Cagliari, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3377325.3377486

IUI '20, March 17–20, 2020, Cagliari, Italy
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7118-6/20/03…$15.00
https://doi.org/10.1145/3377325.3377486

1 INTRODUCTION

Dialog systems (also known as *virtual assistants*, *conversational agents*, *chatbots* or simply *bots*) have attracted considerable attention in recent time. With advances in Natural Language Processing (NLP), deep learning, as well as ubiquitous access to application programming interfaces (APIs) and information sources, the stage is set for an explosion of bots with powerful potential. Almost all of the big companies have already invested in virtual personal assistants. Apple Siri, Microsoft Cortana, Amazon Alexa and Google Assistant, to name a few, are collectively being used by millions of users worldwide [9]. Many other sophisticated bots are being developed, from those that allow data scientists to assemble data analytic pipelines (e.g., AVA [30], Analyza [11]) to bots that act like human (e.g., Microsoft's Tay).

Essentially, building *task-oriented* bots requires processing a given user utterance[1] (e.g., "search for a restaurant near the university") to identify the user's intent (e.g., *business search*) along with its entities[2] (e.g., *business= "restaurant", location= "university")*. "Intent" refers to the user's purpose in an utterance, and "entity" refers to a term in the given utterance that provides a value of a variable of the intent [68]. The success of intent recognition models heavily relies on obtaining large and high-quality corpora of annotated utterances (i.e., training data). An annotated utterance (e.g., "search for a restaurant near the university" where *intent="business search"*, *business="restaurant"*, and *location="university"*) is a user utterance labeled with a specific intent and its corresponding known entities.

Research into the acquisition of training data for bots has flourished lately [6, 32, 57]. Obtaining training data typically involves two main steps: (i) obtaining an initial sentence that captures users' intention, and (ii) paraphrasing this initial sentence into multiple variations [62, 66, 67]. Paraphrasing is necessary since having a *diverse* set of utterances in the training set can better represent the different ways in which people may specify an intent, especially given the ambiguous and flexible nature of the human language [63]. A lack of variations in training samples may result in bots

[1] also referred to as simply *utterance* or *user input*
[2] also known as slots or parameters

making incorrect intent detection or entity resolution, and therefore perform undesirable (even dangerous) tasks (e.g., pressing the accelerator instead of the brake pedal in a car) [25].

Existing solutions for obtaining and paraphrasing training samples involve either *automated* or *crowdsourcing* techniques. Automated paraphrase generation [12, 14, 39] is potentially cost-free. However, existing state-of-art techniques fall short in producing sufficiently diverse and semantically correct paraphrases [2, 21, 24, 65]. Outcomes from crowdsourcing tasks must be checked for quality since they are produced by workers with unknown or varied skills and motivations [10]. For example, spammers, malicious or inexperienced workers can provide misleading and erroneous paraphrases [66]. The lack of high-quality training samples can be disastrous in some cases [45]. For example Microsoft's chatbot *Tay*, which quickly made a number of racist, sexist, and offensive commentaries because of presence of stereotype and offensive word biases in training data [25].

This paper focuses on a specific but important quality issue in crowdsourced paraphrases: the lack of diversity in the user utterances obtained from the crowd (used for training bots). Indeed, research has shown that crowd workers are biased towards the vocabulary and structure used in the initial sentences provided to them when performing the paraphrasing task, thereby negatively impacting the diversity of training utterances [6, 50, 62]. Bias towards vocabulary and structure of the sentence to be paraphrased can be explained by the *priming effect* – an automatic, implicit and non-conscious activation of information in memory [26]. According to the priming effect, exposure to a stimulus affects responses to a subsequent stimulus. (e.g., being asked to name a word starting with "str", humans are more likely to form the word "strong" than "street" if they have previously been shown the word "strong") [64]. As such, primed by the words in the given utterance, crowd workers are more likely to use the same vocabulary when paraphrasing [29, 50]. Thus, the priming effect may negatively impact the diversity of collected paraphrases.

In this work, we hypothesize and demonstrate that recommending words/phrases can positively prime crowd workers to use new vocabularies in their paraphrases. In other words, we leverage the priming effect itself to devise diversity-enhancing paraphrasing techniques, countering the negative impact of priming given by the words inside the given utterance to be paraphrased. Inspired by automated paraphrase generation techniques while also considering priming effects, we propose a novel *hybrid* (automated-crowdsourced) approach that dynamically suggests words/phrases to workers, to assist them in recalling words. The suggestions can potentially improve the diversity of paraphrasing for a given intent. Suggesting words is challenging because we need to ensure that suggestions do not promote semantically invalid paraphrases. For example, if the intent is to *"find a restaurant"*, suggesting words such as "kitchen" and "counter" (while related to "restaurant") would result in generation of a paraphrase such as *"find a counter"* which does not convey the same intent. Therefore, a key challenge of dynamically generating these suggestions is recommending words/phrases that will improve diversity without resulting in semantically invalid paraphrases.

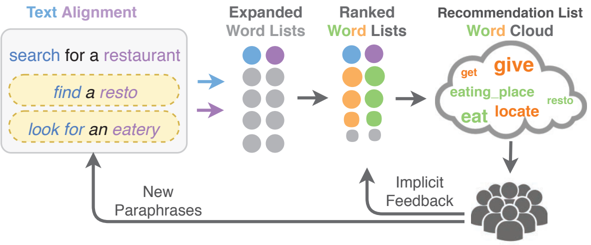


Figure 1: Word-Recommendation Overview

We contribute a novel framework (depicted in Figure 1) combining techniques from dynamic words list expansion [18] and implicit relevance feedback [31] to provide diverse and semantically relevant training utterances for a given intent. More specifically, we make the following contributions:

- **A novel words list expansion technique to improve diversity of training utterances.** We formalize the generation of alternative words/phrases as a problem of expanding a list of seed words/phrases, extracted from an utterance (to be paraphrased) and its previously collected paraphrases. Our approach leverages *word alignment* [5] and *word embedding* [40] techniques. Word alignment is the task of identifying translation relationships between two sentences. For example, word alignment between *"search for a restaurant near the university"* and *"find a restaurant close to the university"* results in the following translation relationships (synonym sets): "search for" ⟶ "find", and "near" ⟶ "close to". Aligning words allows deriving a more accurate sense for each word, since a word may often have multiple meanings depending on its context. For instance, the word *"near"* may also mean *"almost"* and *"approach"* in different sentences. However, using word alignment, we can derive words like *"close to"* to understand the correct word sense. *Word embedding* can then be used to further enrich the synonym sets by mapping words to semantically similar candidates (e.g, *"neighbourhood"*, *"in close proximity"*) [17, 40, 49].
- **A novel probabilistic model to recommend diverse but semantically relevant words**. Unsupervised techniques such as word embedding can generate a large number of related words. Moreover, when combined with techniques such as *word alignment* to help the generation of highly related words list, in many cases the output results may still not be perfect. As a simple example, alternatives that are generated using word embedding for the word *"green"* may include {"greenable", "virescent"}. These words would seemingly be unnatural to be used as paraphrases, albeit a machine would not be able to interpret this. More importantly, obtaining a very large list of suggestions may not necessarily be useful unless we devise methods for ranking (and continuously re-ranking) the results in order to choose and present the set of top-quality words to the crowd worker. To achieve this, we propose a probabilistic model that uses various indicators (e.g., implicit feedback from workers, diversity maximization and semantic relevance) to automatically and adaptively synthesize a reliable score to rank generated word list. This allows the generation of improved versions of suggestions, based on continually monitoring implicit worker feedback, in order to balance diversity and semantic relevance.
- **End-to-End Evaluation.** Finally, we evaluate how these automatically generated suggestions impact the diversity of collected paraphrases on various domains, including 40 intents that are designed

for highly popular APIs (e.g., Yelp, Spotify). Experiments show that our approach improves not only the lexical diversity but also other diversity metrics (PINC[8] and DIV [32]). Moreover, we found out that it reduces task completion time and misspelling errors.

2 RELATED WORK

Paraphrasing. Paraphrasing is the task of expressing the meaning of a fragment of text using different words. It has numerous applications in natural language processing systems such as evaluation of machine translation systems, sentence simplification, automatic plagiarism detection, text summarization, and natural language generation [33, 35, 38]. It is also used in question answering and information retrieval systems to reformulate user's queries [6, 57]. Existing solutions for obtaining and paraphrasing training samples involve either automated or crowdsourcing techniques.

Automatic paraphrasing relies on machine translation techniques (e.g., rule-based, statistical and neural machine translation [12, 27, 39]). While automation is scalable and potentially cheaper, existing automatic paraphrasing approaches produce low-quality paraphrases when they are trained using one specific domain and used in another domain [34]. A key challenge is to produce paraphrases that are semantically equivalent to source sentence [2, 21, 24, 65].

Crowdsourcing has also been investigated extensively to obtain natural language corpora for dialog systems [6, 50, 57, 62]. In crowdsourced paraphrasing, an initial utterance, which is usually provided by experts or generated using templates or generative grammars, is shown as a starting point [50, 57, 62]. For example, to gather utterances for the *business search* intent, assuming that seed values for the *"business"* parameter are *"restaurant"* and *"fruit market"*, the following initial utterances may be generated: *"search for a restaurant"* and *"search for a fruit market."* Then crowdsourcing is used to obtain more paraphrases.

Other research for collecting user utterances focus on obtaining utterances from users through launching a bot [11]. However, launching a machine learning based bot still requires annotated utterances to build an initial version of the bot. In such cases, the quality of the initial set of utterances is of paramount importance because bot users may turn away from the bot if it keeps failing to recognize their intentions.

In our work, we leverage crowdsourced paraphrasing by showing a list of potential word suggestions. We propose an efficient technique to automatically generate a dynamic list of potential alternatives for the words/phrases in a given utterance. Crowdsourcing is then employed by providing workers with the given utterance to paraphrase alongside the dynamic list of alternatives in order to improve diversity.

Diversity in crowdsourced paraphrasing. Human language is rich and a single utterance can be expressed in various ways. Thus, collecting diverse paraphrases is necessary for building robust intent recognition models. However, as a result of being biased towards using words used in the given utterance, crowdsourced paraphrases may lack the desired diversity [29, 29, 46, 47, 50, 54]. Some existing techniques focus on paraphrasing utterances obtained from the crowd instead of initial utterances to increase diversity [46]. Nevertheless, this approach is also prone to producing semantically divergent paraphrases since an invalid paraphrase (to start with) can cascade into other invalid results on subsequent iterations [3, 8, 29]. Another technique for mitigating the priming effect in crowdsourced paraphrases is to replace entities with their images in the initial utterances (e.g., showing a photo of a restaurant instead of using the word "restaurant") [50]. Nevertheless, it is challenging to generalize this approach for non-entity words (e.g., verbs, adjectives, abstract nouns). Likewise, videos have been presented to demonstrate intents in which workers are asked to state the intent in the videos [8]. However, finding/creating such videos is very time-consuming [8].

Other research focused on metrics to measure the diversity of a given set of utterances, namely *Type-Token Ratio (TTR)*, *Paraphrase In N-gram Changes (PINC)* [8], and *Diversity (abbreviated as DIV)* [32]. TTR calculates the rate of unique words to the total number of words. It rewards the use of new words without considering differences in sequences of words in utterances. To address this limitation, PINC measures the percentages of common n-grams between the initial utterance and the rest of utterances without considering inter-paraphrase n-gram changes. DIV calculates n-gram changes between all pairs of collected utterances.

The results of these studies are certainly useful and provide valuable understanding of factors and threats impacting quality in paraphrasing as well as quantifying diversity. We built upon these metrics and propose a novel approach to improve the diversity of collected paraphrases and reduce the number of semantically invalid paraphrases by prioritizing suggestions that are semantically equivalent to the given utterance. Suggesting inappropriate words (e.g., suggesting "kitchen" as a related word for "restaurant") may result in generation of an invalid paraphrase which does not convey the same meaning as the given utterance (e.g., "find a kitchen" for "find a restaurant"). Therefore, a key challenge when dynamically generating these suggestions is recommending words that will improve diversity without resulting in semantically invalid paraphrases. We achieve this by (i) proposing a novel words list expansion technique to generate a list of related words using word-alignment and word embedding techniques (see Section 3.1 & 3.2); and (ii) a novel probabilistic model to balance between promoting diversity and suggesting semantically relevant suggestions by getting implicit feedback from crowd workers to detect noises in the dynamic recommendation list (see Section 3.3).

Priming in crowdsourcing. In experimental psychology, the term *"priming"* refers to a technique in which introducing one stimulus influences a person's response to a subsequent stimulus [26]. In other words, humans may be biased by prior stimuli that affects future processing of information [64]. Word-fragment completion (WFC) is an example of a task where priming may have an impact [55]: a person is given a fragment of a word like "str- - -" and is asked to complete the word. This person is more likely to form the word "strong" than "street" if she had been shown the word "strong" before performing the task [4]. This type of priming is called *repetition priming*. Repetition priming (also called *direct priming*) refers to a form of priming according to which the brain responds more quickly to a stimulus if it has experienced it previously [19].

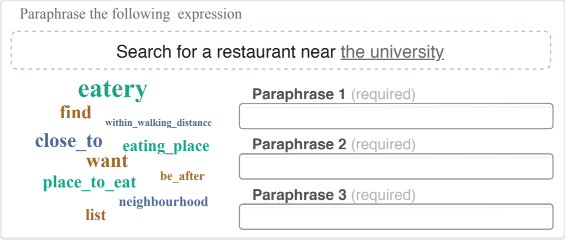
Priming is used in several applications, from swaying public opinion about product marketing, political campaign to sharpening memory skills [1, 13, 22, 61]. Priming has also been used in crowdsourcing to positively affect the performance of crowd workers [20, 23]. For example, research found that showing positive images (e.g., a photo of a smiling child) or listening to positive music when working on tasks can influence idea generation [44]. As another example, priming has been used to trigger workers' inherent motivation to excel in performing a task by showing them quotes of famous figures about "achievement" [20]. In crowdsourced paraphrases, research has also shown that crowd workers are primed by the given utterance as well as by the examples and instructions presented to them. However, in this case, priming negatively impacts diversity since crowd workers are more likely to use the same vocabulary as used in the given utterance and task description [29, 46, 47, 50].

In this paper, we build upon this prior art but also look at priming as an opportunity rather than a problem. Specifically, inspired by tasks such as word-fragment completion, we leverage priming by suggesting a list of words/phrases, that can potentially improve diversity. We hypothesize (and demonstrate) that priming the crowd with a list of potential lexical substitutions can mitigate the negative impact of priming by the words in the given utterance.

Word list expansion. Query rewriting and more specifically query expansion methods can also be adopted to build dynamic word suggestions. The primary goal of query expansion is to solve the term mismatch problem [7] by adding a few words to a given query. In this manner, it aims to mitigate the vocabulary problem when the documents and users use different terminologies to express the same thing. In this paper, we extend a query expansion model based on word embedding [36] by aligning words used in the given utterance and the already collected paraphrases. In this manner, we can distinguish the senses of multi-sense words. For example, word alignment between *"search for a restaurant near the university"* and *"find a restaurant close to the university"* detects the following translation relationships: "search for" ⟶ "find", and "near" ⟶ "close to". Thus, it can be inferred that the word *"near"* conveys a sense meaning *"close to"* (not other senses like *"almost"* or *"approach"*). Furthermore, we adapt the concept of implicit relevance feedback from query expansion methods of information retrieval systems, in order to measure if a word is semantically relevant to the given intent. Relevance feedback in information retrieval systems is used to expand to the query by using contents of relevant documents [53]. However, we use implicit relevance feedback to dynamically remove noises (semantically irrelevant words) from the suggestions. We propose a probabilistic model to measure the likelihood of a word being relevant to the given intent by tracking how suggested words are used. In short, if a word has been suggested multiple times without being used in the collected paraphrases, it is considered semantically irrelevant for the given intent.

3 WORD RECOMMENDER

Figure 2 illustrates an example of a paraphrasing task based on word recommendation on the Figure Eight[3] crowdsourcing platform. The recommendations are shown in a word cloud as a graphical representation. Words inside the word cloud (on the left-hand side) have been generated as relevant words to the given utterance. In the given example, workers have been asked to paraphrase *"search for a restaurant near the university"* which is an utterance for the *business search* intent with *"restaurant"* and *"the university"* as values for *business* and *location* parameters, respectively. Our initial implementation (as presented in Section 4.1) categorizes words with the same color as possible alternatives for one of the words in the utterance. Thereby at the time of paraphrasing, crowd workers are able to pick words from this list and form new paraphrases.



Paraphrase the following expression

Search for a restaurant near the university

Paraphrase 1 (required)

Paraphrase 2 (required)

Paraphrase 3 (required)

Figure 2: Crowd Workers' Interface

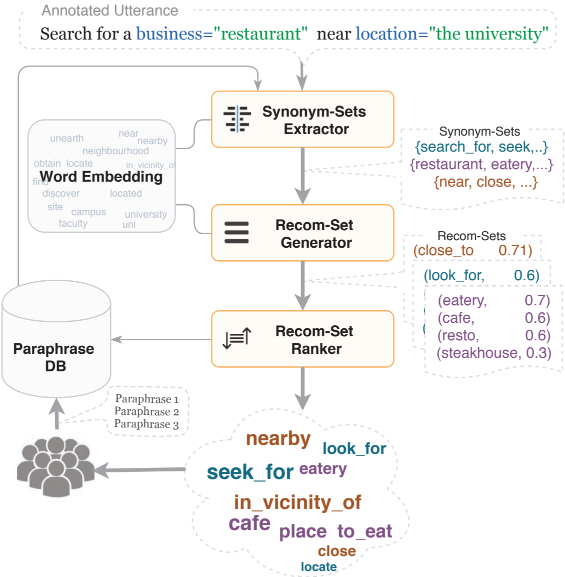


Figure 3: Word Recommender Architecture

The generation of diverse (but relevant) words is done by the *Word Recommender*. Its main components are illustrated in Figure 3. Given an intent (e.g., *business search*) with an annotated utterance (e.g., *Search for a business= "restaurant" near location="the university"*) to be paraphrased, the *Word Recommender* first extracts all words/phrases from the utterance. It then builds a synonym set for each extracted word/phrase by aligning the collected paraphrases with the utterance. Next, for each synonym set, it finds a set of related words to appear in the recommendation list with the help of a word-embedding model. Finally, it (re-)ranks candidate words based on the words appeared in the collected paraphrases to generate a new recommendation list. This process is an ongoing effort: periodically, when workers provide paraphrases for a particular utterance, a new recommendation list is generated according to the collected paraphrases.

[3] https://www.figure-eight.com/

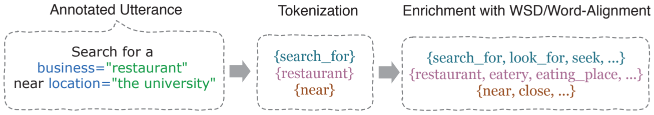

Annotated Utterance
Search for a
business="restaurant"
near location="the university"
Tokenization
{search_for}
{restaurant}
{near}
Enrichment with WSD/Word-Alignment
{search_for, look_for, seek, ...}
{restaurant, eatery, eating_place, ...}
{near, close, ...}

Figure 4: Synonym-Sets Creation for an Utterance

Our system adopts a word embedding model called ConceptNet Numberbatch [56] to find related words for each word appeared in the given utterance. Word embedding methods map words into a vector space model in a way that similar words have similar vectors. As a result, a word embedding model can be used to find similar words by finding the closest neighbors for a given word. ConceptNet Numberbatch has a few characteristics which make it desirable in our work. Firstly, it has been built using retrofitting [17] to refine existing word embedding models with external knowledge bases (e.g., ConceptNet [56].) Secondly, it provides vectors not only for words but also for frequent n-grams up to tri-grams. As a result, using such a word embedding approach, our system is able to suggest both words and phrases. While we have used this model in our experiments, it can easily be substituted with any word embedding model. The rest of this section provides further details about the three main components of *Word Recommender.*

3.1 Synonym Sets Extractor

Any word can have different meanings in different contexts [16]. Since our aim is to suggest highly relevant words to a given utterance, it is important to disambiguate the sense of a given word in a sentence. For example, the word "near" can be an adjective, adverb, or even a verb, and based on its part-of-speech (POS) it can have several synonyms such as *close, approximate, skinny, dear, good*[4]. To suggest appropriate substitutions for such a word, it is important to consider its role and sense in the utterance.

To this end, *Synonym Sets Extractor* first extracts all words/phrases from a given annotated utterance[5], excluding stop-words[6] (since frequent words are less likely to result in repetition priming [58]). We also excluded parameter values marked as *fixed* as a design decision [43] (e.g., {near}, {search_for}, {restaurant}) . Fixing parameter values can be a double-edged sword. Preserving parameter values removes the need for manual annotation of collected paraphrases [43], however it might also reduce variability where we do not want (e.g., cities in the world have many possible values) and allow workers to efficiently focus on where we want variability. On the other hand, if a parameter like "business=restaurant" is fixed then we will be deprived of diverse but relevant utterances such as "where to eat in the university". Best practices for fixing parameters are outside the scope of this paper, and here we simply assume that the bot developer has the choice to do so. Thus, *Synonym Sets Extractor* also ignores words which are marked as fixed.

Next, *Synonym Sets Extractor* enriches each word by creating a set for each word/phrase and adding its synonyms to the set. In a nutshell, *Synonym Sets Extractor* employs (i) a word sense disambiguation (WSD) method if no paraphrase has been collected yet, and (ii) a *word alignment* [5] method to find aligned words of the given utterance and the collected paraphrases. Word alignment is a task of finding the corresponding translation of a word between two pieces of text (e.g., a sentence and its paraphrase). For instance, if we collect a paraphrase like "search for a restaurant in the vicinity of Sydney", we can infer that the word "vicinity" in this context is aligned to "near". In our implementation we used the sentence aligner proposed in [59] to find alternatives for a given word in the collected paraphrases. These alternatives are added to the synonym set of the word to enrich the word and disambiguate its meaning in the utterance.

While word alignment can be used for disambiguation, it is not practical until a few paraphrases are collected from crowd workers. To mitigate this issue when there is no paraphrase at the beginning of the crowdsourcing, the proposed system uses a WSD method proposed in [48]. Using a dictionary such as WordNet [42], the WSD method is able to determine a set of synonyms for a word in the given sentence. Finally, by enriching each word/phrase of the given annotated utterance, the following synonym-sets are obtained: {search_for, look_for, seek}, {restaurant, eatery, eating_place, eating_house}, and {near, close}.

3.2 Recommendation Sets Generator

Recommendation Sets Generator expands a synonym set by adding relevant words to the set, resulting in a *recommendation set.* This component builds upon word embeddings– a method of mapping words into a vector space model in a way that similar words have similar vectors. In particular, it makes use of a word embedding model called ConceptNet Numberbatch [56] to find related words/phrases for each synonym set.

Figure 5 demonstrates how related words are found by *Recommendation Sets Generator.* For a given synonym set (e.g., {near, close}), *Recommendation Sets Generator* calculates its mean vector by averaging the vectors of all words in the synonym set ($\vec{ss} = 1/|ss| \sum_{s \in ss} \vec{s}$; where ss stands for synonym set). Next, it finds the top-n neighbours of the mean vector ($\vec{ss}$) using cosine similarity ($cos(\vec{u}, \vec{v}) = \vec{u}.\vec{v}/||\vec{u}||.||\vec{v}||$), and ranks the neighbours based on their cosine similarity to the vector of the given utterance ($\vec{expr}$)– the average of all content words (nouns, verbs, and adjectives) in the utterance [68]. It should be noted that, while it is possible to order the words/phrases in the candidate sets based on their similarity to the corresponding synonym set, we calculate their similarities to the utterance to give importance to words which are more relevant to the utterance. For example, in Figure 5 *"vicinity"* is ranked higher than *"adjacent"* while the vectors of *"adjacent"* and *"near"* are closer than those of *"vicinity"* and *"near"*; but since *"vicinity"* is closer to the utterance, it is given a higher score.

3.3 Recommendation Set Ranker

The goal of *Word Recommender* is to estimate the probability that a word/phrase will improve diversity of paraphrases if it appears in the recommendation list. Recommending infrequent words can improve the diversity because they enhance the chance of increasing the number of unique n-grams in the collected paraphrases and thus increase the diversity metrics (e.g., PINC[8], DIV [32].) However, recommendations should also be semantically related to the given utterance to avoid the generation of semantically invalid

[4] http://wordnetweb.princeton.edu/perl/webwn?s=near
[5] we assume that utterances are annotated (either manually or automatically)
[6] stop-words are frequently used in text by such as "a", "an", "is"

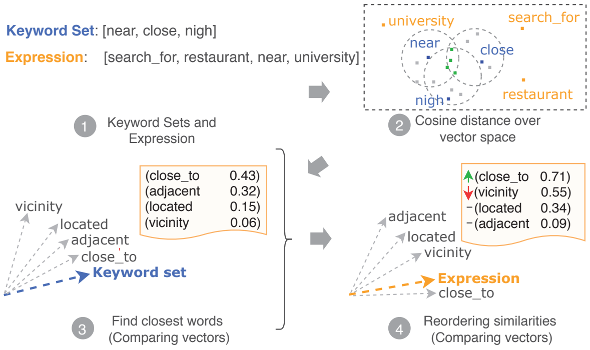


Figure 5: Recommendation-Set Generation

paraphrases. If the recommended word is not semantically related to the given utterance, it will have no effect on the diversity of collected paraphrases because workers may not use them. Or even more dangerously, showing irrelevant recommendations may result in the generation of semantically invalid paraphrases. For example, if the intent is to *"find a restaurant"*, suggesting words like "kitchen" while might be relevant but will result in generating paraphrases which are not specifying the same intent (e.g., "look for a kitchen".)

To this end, we devise a probabilistic model to rank the words in recommendation-sets by estimating the probability that a given word will increase diversity while being semantically related to the intent. To achieve this we propose two probabilistic models for measuring the semantic similarity of a word to the intent, and one probabilistic model for estimating if recommending a word will increase diversity of paraphrases.

3.3.1 Similarity Probability Model (SM). It is important to suggest only words which are highly related to the utterance. This is partly done in the previous step when words in a recommendation-set are scored based on how they are close to the utterance see (Section 3.2). However, to obtain a probability distribution over the recommendation words, we can use softmax-normalization[36]:

$$P(w|\mathcal{SM}) = \frac{exp(cos(\vec{w}, \vec{expr}))}{\sum_{w' \in rs} exp(cos(\vec{w'}, \vec{expr}))} \tag{1}$$

where rs stands for recommendation-set and $w \in cs$. In fact, this probability model, constitutes a unigram language model defined over the recommendation-set assuming that the rest of words have zero probability.

3.3.2 Relevance-Feedback Probability Model (RM). As mentioned before, *recommendation-sets* are noisy and not all of the recommendations can be used in paraphrasing. Our premise is that using an implicit relevance feedback model can assist *Word Recommender* in distinguishing between noises and highly related words. Intuitively, if a word has been recommended but has rarely been used in the collected paraphrases, most likely it is an unrelated word and it should not be present in the next updates. Formally, the maximum likelihood estimate (MLE) of w with respect to the number of times it appeared in recommendation list and used in the paraphrases is:

$$p_{MLE}(w) = \frac{used(w)}{shown(w)}$$

where $shown(.)$ counts how many times a term has been shown to crowd workers, and $used(.)$ counts the number of times a word has been used in the paraphrases. However, this estimation disregards two essential aspects:

(1) Not all of the collected paraphrases are valid, and only valid paraphrases should be considered while counting.
(2) Even if a word is highly relevant to a given utterance, it does not necessarily mean that the worker will include the word in the paraphrases. For example, imagine that there are several synonymous for a word, and all workers pick only a single one of them, and no one uses the rest of synonyms while they might be as appropriate as the chosen one. Such cases can definitely affect the chance of a word to be selected. So we cannot assume that a word which has not been used is completely out of scope.

To address these, let λ be the probability of a given utterance to be valid, and γ be the probability of a word not to be used in a paraphrase regardless of being an appropriate alternative or not; the revised relevance probability is:

$$p_{RMLE}(w) = \frac{\lambda used(w) + \gamma(shown(w) - used(w)) + \epsilon}{shown(w) + \epsilon}$$

In our experiments, we set $\lambda = 0.80$ (base on error rates reported in various works[46, 57]), $\epsilon = 0.1$ and $\gamma = 0.5$ which are initial configurations, and finding the optimal values requires further studies. To avoid zero division errors, we also added a small positive number $\epsilon = 0.1$ to the numerator and denominator. In summary, this function reduces the weights of words which appeared in the recommendation list but were rarely used, as well as words which have been exploited by the paraphrases. The relevance probability can be normalized to obtain the relevance probability distribution under the relevance model (RM):

$$P(w|\mathcal{RM}) = \frac{p_{RMLE}(w)}{\sum_{w' \in cs} p_{RMLE}(w')} \tag{2}$$

3.3.3 Diversity Probability Model (DM). To encourage diversity, infrequent words must be given more priority because they are more likely to generate new n-grams which have not been seen in the collected paraphrases and thus improve diversity metrics. To attend this issue, we used a modified version of BM25 inverse-document-frequency (IDF) [52] to avoid negative numbers:

$$idf(w) = \log(1 + \frac{N - f(w) + 0.5}{f(w) + 0.5}) \tag{3}$$

where $f(.)$ represents the frequency of the word in the collected paraphrases, and N is the total number of the collected paraphrases. The IDF values can be normalized to yield probabilities:

$$P(w|\mathcal{DM}) = \frac{idf(w)}{\sum_{w' \in cs} idf(w')} \tag{4}$$

Finally, we use a linear interpolation technique [41] to approximate to what extent a word increases diversity while preserving semantics:

$$P(w) = \alpha P(w|\mathcal{SM}) + \beta P(w|\mathcal{RM}) + \theta P(w|\mathcal{DM}) \tag{5}$$

where $\alpha, \beta, \theta \in [0, 1]$ are interpolation parameters ($\alpha + \beta + \theta = 1$) and control the trade-off between the probability models. In our experiments, we kept the interpolation parameters equal. Words in each recommendation-set are ranked based on Equation 5; and top-*m* words from each set are selected to be present in the current update of the recommendation list which is shown to workers; where *m* is the rounded value of the size of the list (see Section 4.1) divided by the number of recommendation-sets. In the time of showing the recommendation list, words which belong to the same recommendation set are inked with the same color. Moreover, their final scores determine how big they should appear on the current update of the *word-cloud*.

4 EXPERIMENTS & RESULTS

Before doing a comprehensive experiment, we conducted a pilot to decide on the user interface design of a crowdsourcing task. Based on these observations and interviews, we then evaluated the approach on the Figure-Eight crowdsourcing platform.

4.1 Task Design Experiment

Participants. We recruited a convenience sample of 7 participants including 2 Postdoctoral researchers (P1, P2), 3 PhD students (P3, P4, P5), 1 research assistant (P6) and 1 undergraduate student (P7).

Procedure. The experiment was conducted in the following parts: (a) Firstly, a brief explanation of the task was provided to the participants, as well as a few examples for invalid paraphrases. To not bias participants, we did not give them any valid paraphrasing examples; (b) next, participants were assigned 6 randomly chosen utterances to provide 3 paraphrases for each. Each of participants encountered two utterances with the *word-cloud* size of 10, two with the size of 15, and two with the size of 20; and (c) finally, a follow-up semi-structured interview was conducted about the user experience of using *word-cloud* during the paraphrasing process.

Does *word recommendation* help? One of the aims of this experiment was to know if the automatically generated *word-cloud* helped workers during paraphrasing. All of our participants confirmed that the *word-cloud* assisted them especially for those whose main language was not English:

> *"I think the word cloud can help the users with English as the second language..." (P5)*

> *"I like the idea of giving you some words, it actually helped me specially in those paraphrasing questions that I didn't have any alternatives in mind" (P3)*

> *"Well, yes definitely, specially for me that English is not my mother tongue; I also learned some new words while I was making new sentences for [...]" (P1)*

How many words are appropriate in the recommendation list? Figure 6 demonstrates how participants rated different *word-clouds* (recommendation lists) based on their sizes in a Likert scale with 5 being very appropriate. Generally, most of the participants preferred the size of 10; even it was considered as an accelerating factor by one of the participants:

> *"The more compact ones seemed more useful to me. I guess it is because you don't have to inspect too many options, which allows you to perform the task more quickly." (P2)*

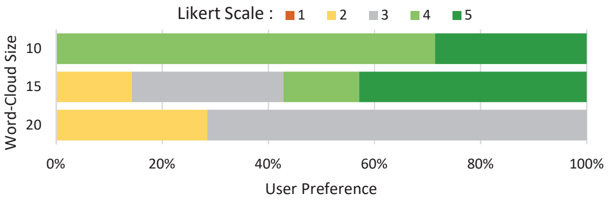


Figure 6: Likert Assessment of Word Clouds by Size

However, a few found the *word-cloud* of size 15 more preferable. Our investigations revealed that the size of *word-cloud* may also be a function of the number of words in an utterance, and for short utterances showing many alternatives for a single word is considered inappropriate and time-consuming sometimes:

> *"...having more words makes it hard to choose." (P4)*

> *"..., also I think 10 is enough, even less let's say 7 or 8; 20 is definitely too much." (P6)*

Therefore, in our follow-up experiments we set a limit of 10 for the total size of the *word-cloud* and only 5 for single word alternatives.

Is coloring similar words with the same color helpful? While some of participants found word coloring very helpful, others did not understand the semantics behind the coloring:

> *"Word coloring is great and should be kept as it is..." (P7)*

> *"But the coloring was confusing. I cannot really assign a specific characteristics to the used colors...while doing the task, the semantics of the colors did not come immediately and naturally to my mind" (P2)*

Based on the interviews, we decided to use coloring with muted colors (avoiding bright ones). To resolve misunderstandings about colors, we also explained the coloring in the task instructions.

4.2 Crowdsourced Paraphrasing

To verify our approach, we randomly selected 40 utterances indexed in API-KG [68] –a knowledge graph designed for RESTful APIs – and ThingPedia [6] including utterances from different domains: Yelp, Skyscanner, Spotify, Scopus, Expedia, Open Weather, Amazon AWS, Gmail, Facebook, and Bing Image Search.

Next, we launched five paraphrasing jobs on Figure-Eight: (i) a simple *baseline* which simply asks crowd workers to paraphrase given paraphrases; (ii) the state-of-art approach named *Chinese Whispers (CW)* [46]; (iii) the proposed approach called *Word Recommender (WR)*; (iv) a recommendation method with words generated by an open-source query rewriting method called *SearchBetter (SB)*[7]; and (v) finally, inspired by the Taboo game, we created another baseline called *Taboo Words (TW)* by forbidding workers from using the words which have high frequencies in the already collected paraphrases; we excluded stopwords and in each round only 5 taboo words were shown to the crowd workers; since higher numbers of taboo words makes the paraphrasing task very difficult[8]. We did not compare our system with approaches like replacing entities

[7] available at https://github.com/hathix/searchbetter; where we combined the two query rewriting methods of SearchBetter (Wikipedia and Word2Vec rewriters) and fed the suggestions into the word cloud in the order given by the framework

[8] we first experimented with 10 taboo words but workers stopped completing the task and rate the difficulty of the task as 1 out of 5

with images or showing videos because they are difficult to adopt in general, as discussed earlier in this paper.

For each of the five jobs, we collected 10 judgments per utterance. In our jobs, a judgment is a triple of paraphrases submitted by a worker. Workers were asked to provide three paraphrases for each utterance to reduce repetitive paraphrases [29] which is a common practice [6, 29] in crowdsourced paraphrasing. Each worker gained 10 cents per judgment, and totally we spent about $258 including the transaction fee charged by the platform. Over a span of 3 days, we collected 30 paraphrases per utterance per approach from English speaking countries, and created five datasets containing 6000 paraphrases in total.

Procedure. First, the workers were instructed to be familiar with the task and its constraints (e.g., parameter values must be used in the paraphrases as they are in the given initial utterance). Next, for a given utterance, participants were asked to provide three paraphrases. In the case of using one of the word recommendation methods, each worker was also asked if the generated recommendations helped them during the task using the Likert scale (1 to 5). In the following section, we discuss different aspects of our experimentation.

Cleaning. After collecting the paraphrases, we launched another crowdsourcing task to qualify crowdsourced paraphrases to determine correct and incorrect paraphrases. To this end, we assigned each paraphrases to 3 workers, where each worker gained 2 cents per annotating triple-paraphrases, and totally we spent about 144 dollars. To further increase the quality of annotations and resolve disagreements between crowd workers, two authors of this paper manually checked, discussed, and revised the labels. As it is also shown in Table 1, the sizes of datasets are roughly equal after pruning, and roughly 20% of the collected paraphrases are incorrect (except the *TW* and *CW* methods). This is also in-line with the value chosen for $\lambda = 0.8$ in Equation 2. As mentioned before, *CW* is prone to producing many incorrect paraphrases [29]). Moreover, based on our observations, the *TW* method makes the task very difficult for crowd workers and as a result, many incorrect paraphrases are generated to circumvent the forbidden words. In the following sections, we compare the diversity of the datasets only based on the correct paraphrases.

Table 1: Crowdsourced Paraphrase Datasets

Dataset	Total Size	Correct	Incorrect
Baseline	1200	935 (78%)	265 (22%)
Chinese Whispers (CW)	1200	823 (69%)	377 (31%)
SearchBetter (SB)	1200	986 (82%)	214 (18%)
Taboo Words (TW)	1200	770 (64%)	430 (36%)
Word Recommender (WR)	1200	974 (81%)	226 (19%)

4.3 Results

Does word recommendation improve diversity? The main aim of the proposed approach is to improve the diversity of collected paraphrases by stimulating users to use words/phrases that can add variety to the paraphrases collected for a given intent. To measure diversity of collected paraphrases, after removing punctuation marks, lowercasing, and lemmatizing paraphrases, we calculated four different measures described in Section 2: (1) TTR, (2) PINC, (3) DIV, and (4) the vocabulary size.

Table 2: Diversity Comparison

Dataset	TTR	PINC	DIV	Vocabulary Size
Baseline	0.258	0.653	0.382	1647
Chinese Whispers (CW)	0.278	0.695	0.365	1622
SearchBetter (SB)	0.285	0.724	0.484†	1713
Taboo Words (TW)	**0.338**†	0.733†	0.518†	1682
Word Recommender (WR)	0.313†	**0.734**†	**0.543**†	**2064**

† indicates two-tailed statistical significance at the 0.01 level over baseline.

Using a two-tailed independent-samples t-test, there was a significant difference in the lexical diversity (TTR) for WR ($M = 0.32, SD = 0.72$) over the baseline ($M = 0.26, SD = 0.05$); $t(38) = 4.01, p = 0.0001$. As shown in Table 2, these results suggest that the WR does have an effect on lexical diversity; on average, using WR enhances lexical diversity by 21.32%. Although TTR is less for longer documents, knowing the fact that both datasets are almost equal in size, we can compare the TTRs. Moreover, WR increased the vocabulary size by 19.92%. By comparing the vocabulary sizes of two datasets, we can also infer that using WR yields more diverse paraphrases. On the other hand, while SB ($M = 0.29, SD = 0.84$) improves TTR over the baseline, it is not statistically significant; $t(38) = 1.82, p = 0.07$. Moreover, even though TW proceeds WR in terms of TTR, these two are not comparable since there is a big difference in the number of paraphrases in the two datasets, making TTR a not very suitable metric for comparing the datasets.

TTR cannot measure how much structurally diverse are two datasets; to overcome this issue, PINC has been introduced. PINC measures the percentages of n-grams in the source sentence and its paraphrase; in short, PINC rewards introducing new n-grams. An independent-samples t-test was also conducted to compare the PINC scores. The PINC values indicated that the paraphrases generated by WR ($M = 0.73, SD = 0.09$) are more diverse than those written using the baseline approach ($M = 0.65, SD = 0.11$); $t(38) = 3.42, p = 0.001$. WR also improves mean average PINC by 12.4%. Using TW ($M = 0.73, SD = 0.09$) also yields statistically significant improvement on PINC; $t(38) = 3.39, p = 0.001$.

One problem with PINC is that it only considers n-gram changes between the source sentence and its paraphrases; without considering that of between two paraphrases. DIV [32] is another diversity measure which overcomes this issue by pair-wide n-gram comparison between paraphrases. Our experiments indicate that WR yields 42.15% improvement over the baseline. Interestingly, CW has a lower DIV than the baseline; one reason behind that may lay in the fact that CW tries to make diverse paraphrases regarding the source utterance, but it fails to promote diversity between paraphrases. The TW approach also improves DIV by 35.6%.

Given the above-mentioned results, we concluded that word recommendation as well as showing taboo words improve not only lexical diversity but also PINC and DIV. However, TW results in generating too many incorrect paraphrases as shown in Section 4.

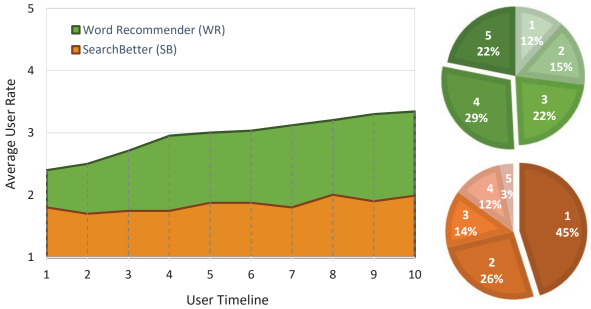


Figure 7: Average User Rating over Time

Does Word Recommender offer appropriate alternatives? To track the quality of our dynamic word recommendation list, we asked crowd-workers to rate how helpful the generated list was for that particular task. Figure 7 illustrates the average user rating of 40 utterances over time (the 1st user to 10th user) for both WR (green) and SB (orange). This figure reveals that the proposed approach for creating the recommendation list and the probability model for re-ranking are properly working and over time improve the quality of the generated list. Figure 7 also shows the overall ratings for the WR (the green pie chart) and SB (the orange pie chart) approaches. As shown in these pie charts, the most of suggestion lists are rated 4 in the WR approaches while that of SB is 1. On the other hand, almost without any trend for improving, the performance of SB fluctuates over time. Since both approaches use the same word-embedding model, we can conclude that WR outperforms the SB mostly because of the probability model described in Section 3.3 by reducing noises over time while trying to propose new words. As opposed to WR, SB tries to exploit new words. Moreover, Figure 7 indicates that crowd workers are more happy with WR's dynamic lists than those of SB.

Does word recommendation impact the semantic error rate? While improving diversity is of paramount importance, it is also essential for any approach not to advocate collecting semantically invalid paraphrases. To assure that our proposed approach does not increase semantic error rate[9], we compared the datasets. In this comparison we deducted the paraphrases which have spelling errors from the total number of incorrect paraphrases, assuming that the rest incorrect paraphrases semantically differ from the original utterance. We noticed that the datasets created by the word recommendations approaches (SB and WR) have fewer semantically incorrect paraphrases. Thus it can be concluded that using word recommendation does not increase the number of semantically incorrect paraphrases in comparison to the rest of the approaches. On the other hand, while TW generates diverse paraphrases, it has the most number of semantically incorrect paraphrases. Based on our observations, it lies in the fact that forbidding users from using words makes the task very difficult to perform and many of workers started to generate garbage paraphrases to accelerate the paraphrasing process.

Does word recommendation impact naturalness? We refer naturalness as the likelihood of an utterance occurring in a real situation. To measure how naturalness is affected by each of the crowdsourcing methods, we randomly selected 100 utterances from each of the methods. Next, we launched a crowdsourcing task on Figure-Eight. Crowd workers were asked to rate how likely a given utterance might happen in a real conversation in a scale of 1 to 5 by 5 being highly likely. Totally we spent 21 dollars for the annotation task. Table 3 gives the average naturalness score given by crowd workers. As it is shown in the table, all methods almost perform alike; however, baseline followed by the proposed method surpasses the rest. Given that the difference is not significant, it can be concluded that word recommendation does not significantly impact the naturalness of paraphrases.

Table 3: Naturalness Comparison

Dataset	Naturalness
Baseline	**3.63**
Chinese Whispers (CW)	3.50
SearchBetter (SB)	3.37
Taboo Words (TW)	3.31
Word Recommender (WR)	3.57

Does higher diversity improve bots' performance? The main reason for improving the diversity of user utterances is to build a more accurate bot. To compare the performance of bots built by each method, we used the wit.ai[10] platform and built a bot per API (Yelp, Skyscanner, Spotify, Scopus, Expedia, Open Weather, Amazon AWS, Gmail, Facebook, and Bing Image Search) for each crowdsourcing method (Baseline, CW, SB, WR, and TW). Each bot is trained using the crowdsourced utterances by a particular method, excluding incorrect paraphrases. Next, we evaluated the trained bot against the correct utterances in other datasets in the absence of a gold dataset containing a list of real user utterances. Table 4 shows the average accuracy for intent detection per bot. The bots which were trained on the dataset obtained by the proposed method yield 35% accuracy improvement over the baseline dataset on average. Therefore, it can be inferred that diversity plays a role in the accuracy of intent detection in bot development platforms.

Comparing the average accuracy of each bot and their diversity measures, we recognized that DIV is more in-line with the bot's intent detection accuracy. As can be seen in Table 4, the bot trained on the CW dataset has the lowest accuracy, as it has lowest DIV value as well, while it outperforms the baseline in terms of TTR and PINC.

Table 4: Intent Detection Accuracy by Dataset

Dataset	Accuracy
Baseline	0.619
Chinese Whispers (CW)	0.582
SearchBetter (SB)	0.795†
Taboo Words (TW)	0.771†
Word Recommender (WR)	**0.835**†

† indicates two-tailed statistical significance at the 0.01 level over baseline.

[9] percentage of semantically incorrect paraphrases

[10] https://wit.ai

Does word recommendation reduce spelling errors? To count spelling errors, we used Google Docs editor[11], and manually counted the errors identified by the editor. We observed that the baseline, CW, SB, WR, TW datasets have 29, 23, 11, 16, and 38 spelling errors. The reason behind such a reduction when using a word recommendation based approach (WR and SB) might lie in the fact that workers are less prone to making spelling errors when they have given spellings of words they may use.

Does word recommendation reduces the task completion time? Task completion time indicates how long it takes for a worker to finish the task. Since the time calculated by platforms cannot consider the time a worker spend on unrelated jobs (e.g., talking on phones, having coffees), we calculated the interquartile mean (IQM) for all datasets. The IQM values for the baseline, CW, SB, TW, and WR were 47, 41, 40, 65, and 35 seconds per paraphrase. Therefore, it can be inferred that using the proposed approach can slightly accelerate paraphrasing. It is also in-line with the priming effect that an appropriate set of words/phrases recommendations can help workers to retrieve related words faster. The proposed approach has the minimum completion time among other approaches, while the recommendation list generated by SB shows only a slight improvement in task completion time which might be due to the low-quality of recommendations in comparison to the proposed approach. On the other hand, forbidding workers from using taboo words increases the difficulty of the task, making the task completion time longer.

Does word recommendation increase workers' satisfaction? Upon completion of the task, crowd workers were prompted to take a satisfaction survey for a feedback about various aspects of the task; including for how easy workers found the crowdsourcing job and how satisfied they are regarding the payment they received for doing triple paraphrasing. Table 5 gives the average scores (scaling from 0-5) given by crowd workers for each of the paraphrasing tasks. Based on the scores reported by Figure-Eight, workers found the word-recommendation approaches (SB and WR) easier and fairer regarding the payment. This indicates that recommending words facilitate paraphrasing. On the other hand, workers found the TW approach comparatively difficult and unfair.

Table 5: Worker Satisfaction

Dataset	Ease of Job	Pay
Baseline	3.98	3.66
Chinese Whispers (CW)	4.06	3.75
SearchBetter (SB)	4.47	4.20
Taboo Words (TW)	3.30	3.50
Word Recommender (WR)	**4.62**	**4.64**

4.4 Limitations

Word-recommendation facilitates paraphrasing and it improves diversity of collected paraphrases. However, it is not immune to unqualified crowd workers. Cheaters and unqualified workers generated incorrect paraphrases which can affect the quality of the recommendations [66]. While in the design of the ranking probability model we have taken invalid paraphrases into account, they can still harm the recommendations. This can be mitigated by automatic quality control to only let qualified paraphrases be submitted [66]. This requires automatic detection of incorrect paraphrases, to only allow workers to submit high quality paraphrases. As such, noises can be reduced and consequently quality of word suggestions can be improved.

Another limitation of the proposed system is for the cases in which given words do not have many closely related words. However, we have used a fixed number of top-*n* neighbours for all words as mentioned in Section 3.2. Choosing a proper value for *n* is debatable and depends on how a word embedding model is trained (e.g., its vector space dimension) [15, 51]. As future work, it is essential to dynamically determine the value of *n* for a given synonym-set. Moreover, using domain specific word embeddings can help denoising word suggestions. As such, in a given domain, the proposed system can suggest highly related words and synonyms (e.g., in programming domain, the word "Java" refers to a programming language not to "coffee" or "mocha").

5 CONCLUSION

In this paper, we showed how *word recommendations* can accelerate paraphrasing and improve diversity. We proposed a novel hybrid technique that combines existing advances using both automated methods and crowdsourcing. Our work aimed at addressing an important shortcoming in current crowdsourced paraphrases, namely the *priming effect*. By recommending appropriate words we sought to motivate crowd workers to enhance diversity of their paraphrases. Our solution involved automated methods for selecting seed words and performing word expansion. Nevertheless, the main challenge is to recommend diverse but semantically relevant words. We thus devised a probabilistic model to continuously adapt the expanded list into an improved version; we relied specifically on monitoring implicit worker feedback. Ultimately, our end-to-end experiments indicate that the proposed method improved the diversity of paraphrases.

We observed that a major quality issue with serious effects is malicious workers who generated garbage paraphrases [28, 37, 60]. While we accounted for this problem in the proposed model by the implicit relevance feedback, it can still hurt the performance of recommendation. Our future work will focus on automatically detecting invalid paraphrases. Another important aspect of crowdsourcing is to formalize and define when enough paraphrases have been collected for a given intent. Doing so is not easy [32], and it might depend on the intent detection algorithm, desired accuracy, and many other factors. In future work, we will also target this problem, together with many other exciting opportunities as extensions to this work.

ACKNOWLEDGEMENTS

This research was supported fully by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP1601104515).

[11] https://docs.google.com

REFERENCES

[1] David M Boush. 1993. How advertising slogans can prime evaluations of brand extensions. *Psychology & Marketing* 10, 1 (1993), 67–78.

[2] Florin Brad and Traian Rebedea. 2017. Neural Paraphrase Generation using Transfer Learning. In *Proceedings of the 10th International Conference on Natural Language Generation*. 257–261.

[3] Patricia Braunger, Hansjörg Hofmann, Steffen Werner, and Maria Schmidt. 2016. A Comparative Analysis of Crowdsourced Natural Language Corpora for Spoken Dialog Systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, 750–755. https://www.aclweb.org/anthology/L16-1119

[4] A.M. Brickman and Yaakov Stern. 2010. Aging and Memory in Humans. *Sage Encyclopedia of Neuroscience* 1 (01 2010), 175–180. https://doi.org/10.1016/B978-008045046-9.00745-2

[5] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Comput. Linguist.* 19, 2 (June 1993), 263–311. http://dl.acm.org/citation.cfm?id=972470.972474

[6] Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S. Lam. 2017. Almond: The Architecture of an Open, Crowdsourced, Privacy-Preserving, Programmable Virtual Assistant. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 341–350. https://doi.org/10.1145/3038912.3052562

[7] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.* 44, 1, Article 1, 50 pages. https://doi.org/10.1145/2071389.2071390

[8] David L. Chen and William B. Dolan. 2011. Collecting Highly Parallel Data for Paraphrase Evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 190–200. http://dl.acm.org/citation.cfm?id=2002472.2002497

[9] Robert Dale. 2016. The return of the chatbots. *Natural Language Engineering* 22, 5 (2016), 811–817.

[10] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Comput. Surv.* 51, 1, Article 7, 40 pages. https://doi.org/10.1145/3148148

[11] Kedar Dhamdhere, Kevin S. McCurley, Ralfi Nahmias, Mukund Sundararajan, and Qiqi Yan. 2017. Analyza: Exploring Data with Conversation. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces (IUI '17)*. ACM, New York, NY, USA, 493–504. https://doi.org/10.1145/3025171.3025227

[12] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to Paraphrase for Question Answering. *arXiv preprint arXiv:1708.06022* (2017).

[13] Todd Donovan, Caroline J Tolbert, and Daniel A Smith. 2008. Priming presidential votes by direct democracy. *The Journal of Politics* 70, 4 (2008), 1217–1231.

[14] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question Generation for Question Answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 866–874. https://doi.org/10.18653/v1/D17-1090

[15] Ábel Elekes, Martin Schäler, and Klemens Böhm. 2017. On the various semantics of similarity in word embedding models. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*. IEEE Press, 139–148.

[16] Katrin Erk and Sebastian Padó. 2008. A Structured Vector Space Model for Word Meaning in Context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 897–906. http://dl.acm.org/citation.cfm?id=1613715.1613831

[17] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166* (2014).

[18] Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding Topic Signals in Large-Scale Text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4647–4657. https://doi.org/10.1145/2858036.2858535

[19] Kenneth I Forster and Chris Davis. 1984. Repetition priming and frequency attenuation in lexical access. *Journal of experimental psychology: Learning, Memory, and Cognition* 10, 4 (1984), 680.

[20] Ujwal Gadiraju and Stefan Dietze. 2017. Improving Learning Through Achievement Priming in Crowdsourced Information Finding Microtasks. In *Proceedings of the Seventh International Learning Analytics &#38; Knowledge Conference (LAK '17)*. ACM, New York, NY, USA, 105–114. https://doi.org/10.1145/3027385.3027402

[21] Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A Deep Generative Framework for Paraphrase Generation. *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[22] Jennifer L Harris, John A Bargh, and Kelly D Brownell. 2009. Priming effects of television food advertising on eating behavior. *Health psychology* 28, 4 (2009), 404.

[23] Lane Harrison, Drew Skau, Steven Franconeri, Aidong Lu, and Remco Chang. 2013. Influencing Visual Judgment Through Affective Priming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2949–2958. https://doi.org/10.1145/2470654.2481410

[24] Sadid A Hasan, Bo Liu, Joey Liu, Ashequl Qadir, Kathy Lee, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. 2016. Neural clinical paraphrase generation with attention. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*. 42–53.

[25] Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. 2018. Ethical Challenges in Data-Driven Dialogue Systems. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society (AIES '18)*. 123–129.

[26] Dirk Hermans, Jan De Houwer, and Paul Eelen. 1994. The affective priming effect: Automatic activation of evaluative information in memory. *Cognition & Emotion* 8, 6 (1994), 515–533.

[27] Shaohan Huang, Yu Wu, Furu Wei, and Ming Zhou. 2018. Dictionary-Guided Editing Networks for Paraphrase Generation. *CoRR* (2018).

[28] Ting-Hao Kenneth Huang, Walter S Lasecki, Amos Azaria, and Jeffrey P Bigham. 2016. "Is There Anything Else I Can Help You With?" Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

[29] Youxuan Jiang, Jonathan K. Kummerfeld, and Walter S. Lasecki. 2017. Understanding Task Design Trade-offs in Crowdsourced Paraphrase Collection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 103–109. https://doi.org/10.18653/v1/P17-2017

[30] Rogers Jeffrey Leo John, Navneet Potti, and Jignesh M Patel. 2017. Ava: From Data to Insights Through Conversations.. In *CIDR*.

[31] Seikyung Jung, Jonathan L. Herlocker, and Janet Webster. 2007. Click Data As Implicit Relevance Feedback in Web Search. *Inf. Process. Manage.* 43, 3, 791–807. https://doi.org/10.1016/j.ipm.2006.07.021

[32] Yiping Kang, Yunqi Zhang, Jonathan K. Kummerfeld, Lingjia Tang, and Jason Mars. 2018. Data Collection for Dialogue System: A Startup Perspective. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics, New Orleans - Louisiana, 33–40. https://doi.org/10.18653/v1/N18-3005

[33] David Kauchak and Regina Barzilay. 2006. Paraphrasing for Automatic Evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. Association for Computational Linguistics, New York City, USA, 455–462. https://www.aclweb.org/anthology/N06-1058

[34] Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872* (2017).

[35] Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification Using Paraphrases and Context-Based Lexical Substitution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 207–217. https://doi.org/10.18653/v1/N18-1019

[36] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query Expansion Using Word Embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 1929–1932. https://doi.org/10.1145/2983323.2983876

[37] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J Franklin. 2016. Crowdsourced data management: A survey. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (2016), 2296–2319.

[38] Nitin Madnani and Bonnie J. Dorr. 2010. Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Computational Linguistics* 36, 3 (2010), 341–387. https://doi.org/10.1162/coli_a_00002

[39] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 881–893.

[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. Curran Associates Inc., USA, 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959

[41] David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A Hidden Markov Model Information Retrieval System. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM, New York, NY, USA, 214–221. https://doi.org/10.1145/312624.312680

[42] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11, 39–41. https://doi.org/10.1145/219717.219748

[43] Margaret Mitchell, Dan Bohus, and Ece Kamar. 2014. Crowdsourcing Language Generation Templates for Dialogue Systems. (June 2014), 172–180. https://doi.org/10.3115/v1/W14-5003

[44] Robert R Morris, Mira Dontcheva, and Elizabeth M Gerber. 2012. Priming for better performance in microtask crowdsourcing environments. *IEEE Internet Computing* 16, 5 (2012), 13–19.

[45] Gina Neff and Peter Nagy. 2016. Automation, algorithms, and politics| talking to bots: symbiotic agency and the case of tay. *International Journal of Communication* 10 (2016), 17.

[46] Matteo Negri, Yashar Mehdad, Alessandro Marchetti, Danilo Giampiccolo, and Luisa Bentivogli. 2012. Chinese Whispers: Cooperative Paraphrase Acquisition. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey, 2659–2665. http://www.lrec-conf.org/proceedings/lrec2012/pdf/772_Paper.pdf

[47] Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing nlg data: Pictures elicit better data. *arXiv preprint arXiv:1608.00339* (2016).

[48] Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. 2016. Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, 174–183. https://doi.org/10.18653/v1/W16-1620

[49] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[50] Abhilasha Ravichander, Thomas Manzini, Matthias Grabmair, Graham Neubig, Jonathan Francis, and Eric Nyberg. 2017. How Would You Say It? Eliciting Lexically Diverse Dialogue for Supervised Semantic Parsing. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 374–383.

[51] Navid Rekabsaz, Mihai Lupu, and Allan Hanbury. 2016. Uncertainty in neural network word embedding: Exploration of threshold for similarity. *arXiv preprint arXiv:1606.06086* (2016).

[52] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[53] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press.

[54] Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. Association for Computational Linguistics, New Orleans - Louisiana, 41–51. https://doi.org/10.18653/v1/N18-3006

[55] María J Soler, Carmen Dasí, and Juan C Ruiz. 2015. Priming in word stem completion tasks: comparison with previous results in word fragment completion tasks. *Frontiers in psychology* 6 (2015), 1172.

[56] Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. (2017), 4444–4451 pages. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

[57] Yu Su, Ahmed Hassan Awadallah, Madian Khabsa, Patrick Pantel, Michael Gamon, and Mark Encarnacion. 2017. Building Natural Language Interfaces to Web APIs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. ACM, New York, NY, USA, 177–186. https://doi.org/10.1145/3132847.3133009

[58] Karen Sullivan. 2015. If you study a word do you use it more often? Lexical repetition priming in a corpus of Natural Semantic Metalanguage publications. *Corpora* 10, 3 (2015), 277–290.

[59] Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics* 2 (2014), 219–230.

[60] Martin Tschirsich and Gerold Hintz. 2013. Leveraging crowdsourcing for paraphrase recognition. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. 205–213.

[61] Endel Tulving and Daniel L Schacter. 1990. Priming and human memory systems. *Science* 247, 4940 (1990), 301–306.

[62] W. Y. Wang, D. Bohus, E. Kamar, and E. Horvitz. 2012. Crowdsourcing the acquisition of natural language corpora: Methods and observations. In *2012 IEEE Spoken Language Technology Workshop (SLT)*. 73–78. https://doi.org/10.1109/SLT.2012.6424200

[63] Thomas Wasow, Amy Perfors, and David Beaver. 2005. The puzzle of ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe* (2005), 265–282.

[64] Diana S Woodruff-Pak. 1993. Eyeblink classical conditioning in HM: delay and trace paradigms. *Behavioral neuroscience* 107, 6 (1993), 911.

[65] Qiongkai Xu, Juyan Zhang, Lizhen Qu, Lexing Xie, and Richard Nock. 2018. D-PAGE: Diverse Paraphrase Generation. *CoRR* abs/1808.04364 (2018).

[66] Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Moshe Chai Barukh, and Shayan Zamanirad. 2019. A Study of Incorrect Paraphrases in Crowdsourced User Utterances. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 295–306. https://doi.org/10.18653/v1/N19-1026

[67] Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. 2018. Leveraging Crowdsourcing Data for Deep Active Learning An Application: Learning Intents in Alexa. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 23–32. https://doi.org/10.1145/3178876.3186033

[68] Shayan Zamanirad, Boualem Benatallah, Moshe Chai Barukh, Fabio Casati, and Carlos Rodriguez. 2017. Programming bots by synthesizing natural language expressions into API invocations. In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. IEEE Press, 832–837.