Utterance Retrieval based on Recurrent Surface Text Patterns

Guillaume Dubuisson Duplessis, Franck Charras, Vincent Letard, Anne-Laure Ligozat, Sophie Rosset

Utterance Retrieval based on Recurrent Surface Text Patterns

Guillaume Dubuisson Duplessis[1], Franck Charras[1], Vincent Letard[2], Anne-Laure Ligozat[3], and Sophie Rosset[1]

[1] LIMSI, CNRS, Université Paris-Saclay, F-91405 Orsay
[2] LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, F-91405 Orsay
[3] LIMSI, CNRS, ENSIIE, Université Paris-Saclay, F-91405 Orsay
{gdubuisson, charras, letard, annlor, rosset}@limsi.fr

Abstract. This paper investigates the use of recurrent surface text patterns to represent and index open-domain dialogue utterances for a retrieval system that can be embedded in a conversational agent. This approach involves both the building of a database of such patterns by mining a corpus of written dialogic interactions, and the exploitation of this database in a generalised vector space model for utterance retrieval. It is a corpus-based, unsupervised, parameterless and language-independent process. Our study indicates that the proposed model performs objectively well comparatively to other retrieval models on a task of selection of dialogue examples derived from a large corpus of written dialogues.

Keywords: Dialogue utterance retrieval; Example-based dialogue modelling; Open-domain dialogue system; Evaluation

1 Introduction

Conversational systems are recently gaining a renewed attention in the research community, 50 years after the famous ELIZA system [18], as shown by the recent effort to generate and collect data from the (RE-)WOCHAT workshops [4]. This renewed attention is motivated by the opportunity of exploiting large amount of dialogue data to automatically author a dialogue strategy that can be used in conversational systems such as chatbots [2, 3].

In this paper, we consider the task of automatically authoring an open-domain conversational strategy from unlabelled dialogue data. The main goal is to provide a dialogue system with the ability to appropriately react to a large variety of unexpected out-of-domain human utterances in order to offer an engaging continuation to the dialogue. In this direction, approaches under study can be broken down into generation-based approaches that aim at creating a response given a conversational context (e.g., [17]), and selection-based approaches that focus on selecting a response from a large set of utterances (e.g., [2, 5, 3]). This work focuses on the selection-based approach. More specifically, we view

[4] See http://workshop.colips.org/re-wochat/ and http://workshop.colips.org/wochat/

the problem as an instance of example-based dialogue modelling [8] where the goal is to rank dialogue examples from a large database in order to retrieve the best one. In this work, we are interested in the specific case where a dialogue example is an initiative (I)/response (R) pair (e.g., "(I) do you like paella? (R) yes, it's delicious."). The task aims at retrieving a dialogue pair given an input utterance in a large database of examples. The main idea is to rank initiative utterances from the database of examples against the input utterance to determine the dialogue example that fits best. In this paper, we propose to consider patterns of language use – occurring in a social, opportunistic and dynamic activity such as dialogue – to compare utterances. Our approach can be viewed as an instance of sequential pattern mining [11] applied to information retrieval in textual dialogues. The main contributions of this work are: (i) the extraction of recurrent surface text patterns (RSTP) from a corpus of written utterances; (ii) the representation of utterances as a bag-of-RSTPs; and (iii) the similarity measure between utterances that both takes into account the inverse frequency (IDF) of RSTPs and the relatedness between two RSTPs based on the Jaccard index. We assess this model on a task of utterance selection and show that it outperforms standard models.

Section 2 discusses related work. Section 3 describes the proposed model based on recurrent surface text patterns and outlines its main features. Next, Section 4 describes the adopted experimentation protocol along with the database of dialogue examples created in this work. Then, Section 5 presents and discusses the main results. Finally, Section 6 concludes this paper.

2 Related Work

Several retrieval models have been explored to select the most appropriate dialogue example from the database. The most common ones are vector-space models at the token level along with the cosine similarity [2] and classic Term Frequency-Inverse Document Frequency (TF-IDF) retrieval models [5, 3]. This has also been framed as a multi-class classification problem, e.g., resolved with a perceptron model [5]. More recently, recurrent neural networks have also been proposed to predict if an utterance r is a response associated to a context c formed by a sequence of words [10]. Retrieving an appropriate utterance may also be considered as a short text retrieval problem, the query being the user initiative. From this point of view, the problem is close for example to a community question answering (cQA) problem [12], which aims at finding the existing questions that are semantically equivalent or relevant to the queried questions. Yet, contrary to the cQA problem, the surface form is at least as important as the semantic correspondence between the initiatives, and the objective is not necessarily to give relevant information, but to keep the user engaged in the conversation. Our approach aims at exploiting the recurrent surface text patterns of language use appearing across utterances to represent, index and efficiently retrieves similar utterances in a large database. Its main features are to implement a corpus-based, unsupervised, parameterless and language-independent process.

3 Recurrent Surface Text Pattern-based Approach

In this work, we present a corpus-based process which aims at representing and indexing utterances for a retrieval system. This process is based on two main steps: (i) the building of a database of recurrent surface text patterns by mining a corpus of written dialogic interactions; and (ii) the exploitation of this database in a generalised vector-space model for utterance retrieval.

3.1 Mining of Recurrent Surface Text Patterns (RSTP)

An utterance is viewed as a sequence of tokens. For instance, the utterance "how do you usually introduce yourself ?" (u_1) involves 7 tokens. Similarly, the utterance "how do you know ?" (u_2) contains 5 tokens. We define a recurrent surface text pattern (RSTP) as being a contiguous sequence of tokens that appears in at least two utterances. For example, "how do you" is a RSTP appearing both in utterance u_1 and u_2. However, u_1 and utterance "hi !" do not share any RSTP. Intuitively, RSTPs are surface patterns of language use appearing across utterances in dialogue.

RSTPs are mined from a corpus to form a database further used to represent seen and unseen utterances. Our approach is an instance of sequential pattern mining [11]. The mining process consists in resolving the multiple common subsequence problem by using a generalised suffix tree [6] (resolution of this problem is usually performed to find common substrings in biological strings such as DNA, RNA or protein). Each utterance of the corpus is represented as a sequence of tokens. Let say we have K utterances which lengths sum to N (i.e., the corpus contains N tokens). Each utterance is inserted in the generalised suffix tree. Then, the tree is used to find the subsequences common to k utterances with k ranging from 2 to K. Each node in the tree keeps track of the number of utterances containing the subsequence in the corpus. Remarkably, this problem can be solved in linear time $O(N)$ where N is the total number of tokens in the corpus [6]. Before insertion, utterances are added special begin and end markers (noted, respectively, #B and #E). These markers allow to represent RSTPs starting or ending an utterance. For instance, the subsequence "#B how do you" is a RSTP of u_1 and u_2. However, a single marker is not considered as a RSTP (begin and end markers are excluded from 1-token RSTP).

RSTPs and the standard n-gram model both consider subsequences of tokens. However, they are not to be confused. Indeed, RSTPs belonging to a set of utterances are a subset of all the possible n-grams of this utterance set (with n varying from 1 to the maximum utterance length in the set). However, one important feature of a RSTP is to be recurrent. It means that it must appear in at least two utterances of a corpus (this is not necessary for a n-gram). Last but not least, a RSTP is not limited in size while a n-gram is by definition a contiguous sequence of n items. This work further empirically shows in section 4.2 that the number of unique RSTPs in a corpus of around 3 million of utterances is comparable to the number of unique 3-grams.

3.2 RSTP-based Model

From Vector Space Model to Generalised Vector Space Model The vector space model (VSM) [15] has been widely adopted in information retrieval to determine the relevance of a document to a query. It relies on a set of terms t_i $(1 \leq i \leq n)$ used to indexed a large amount of documents d_α $(1 \leq \alpha \leq p)$. This model assumes that it exists a set of pairwise orthogonal term vectors $\boldsymbol{t}_i$ $(1 \leq i \leq n)$ corresponding to the indexing terms. This set is assumed to be the generating set of the vector space. This vector space is then used to represent as linear combinations of the term vectors both the documents $\boldsymbol{d}_\alpha = \sum_{i=1}^{n} a_{\alpha i}\boldsymbol{t}_i$ and the query $\boldsymbol{q} = \sum_{j=1}^{n} q_j \boldsymbol{t}_j$. The similarity between a document and a query is based on their scalar product which is given in Equation 1.

$$\boldsymbol{d}_\alpha \cdot \boldsymbol{q} = \sum_{i=1}^{n} a_{\alpha i} q_i \qquad (1) \qquad\qquad \boldsymbol{d}_\alpha \cdot \boldsymbol{q} = \sum_{j=1}^{n}\sum_{i=1}^{n} a_{\alpha i} q_j \boldsymbol{t}_i \cdot \boldsymbol{t}_j \qquad (2)$$

A standard retrieval strategy is to rank documents according to their similarity to the query (e.g., the cosine similarity). However, the orthogonality assumption of the VSM is often viewed as being too restrictive and unrealistic. Indeed, it does not take into account the relatedness between pair of terms whereas it might be argued that terms often relate to each other. The generalised vector space model (GVSM) has been proposed to incorporate a measure of similarity between terms into the retrieval process [19]. In doing so, it removes the pairwise orthogonality assumption. The similarity between a document and a query is based on the generalisation of the scalar product given in Equation (2), which also is a measure of their similarity between two normalised vectors (the cosine similarity). Notably, if pairwise orthogonality is assumed, Equation 2 becomes Equation 1. To rank the documents, it is required to know (i) the components $a_{\alpha i}$ and q_j along the term vectors, and (ii) the similarity between every pair of term vectors expressed by $\boldsymbol{t}_i \cdot \boldsymbol{t}_j$ (the explicit representation of term vectors $\boldsymbol{t}_i$ is not required).

Representation of Utterances We model utterances by a GVSM where terms are RSTPs. Utterances are represented by a bag of the most representative RSTPs they include. A RSTP r is representative of an utterance if it does not exist another RSTP r' included in the utterance such that r is a subsequence of r'. Formally, let R be the set of all RSTPs included in an utterance u. $r \in R$ is a representative RSTP of u iff $r \in R$ and $\forall r' \in R$, $r' \neq r$, $r \not\subset r'$. For example, let say we have a RSTP database D = {"how","you know","? #E", "#B how", "#B how do you","#B Hi ! #E"}. The RSTPs included in u_2="how do you know ?" are: R = {"how", "you know", "? #E", "#B how", "#B how do you"}. And the final representation keeping only the most representative RSTPs is: {"you know", "? #E", "#B how do you"}.

This representation ensures that there is not two RSTPs r and r' indexing an utterance such that $r \subset r'$. A particular case of this representation is a

recurrent utterance (i.e. appearing several times in the corpus). In this case, the utterance is a RSTP and is thus represented by itself. In this work, we empirically show in section 4.2 that this representation is sparse. One advantage of this representation is that it takes into account the word order to the extent of patterns (contrary, e.g., to a unigram model). Another one is that RSTPs are easily understandable from a human perspective.

In practical terms, finding RSTPs included in an utterance from a large database can be costly for a real-time interaction system if done naively. The first way is to search whether a RSTP is included in the utterance by taking each one of the RSTP in the database. This way can quickly become impractical if the database is very large. Another way consists in considering all the subsequences of the utterance and test whether this subsequence is a RSTP. This way is often more efficient because of the small size of utterances (some recent work reports that the maximum size of utterances is less than 30 tokens [3]).

Retrieval Strategy The retrieval strategy takes into account relatedness between pairs of terms because RSTPs may be closely related (e.g., "#B how" and "#B how do you"). Similarity between two RSTPs is based on the following idea: the more the sequence of tokens of two RSTPs are similar, the more the RSTPs are similar. Conversely, two RSTPs are said to be orthogonal if they do not share a subsequence of tokens. Formally, we estimate $\boldsymbol{t}_i \cdot \boldsymbol{t}_j$ by a variant of the Jaccard index:

$$\frac{|lgcs(t_i, t_j)|}{|t_i| + |t_j| - |lgcs(t_i, t_j)|}$$

where $|lgcs(t_i, t_j)|$ is the size of the longest common subsequence between t_i and t_j. $\boldsymbol{t}_i \cdot \boldsymbol{t}_j$ is 0 when t_i and t_j do not share any token while it is 1 when $i = j$. Similarity between two utterances is given by Equation 2. Let W_i be the weight assigned to RSTP t_i (the components of the vector). It is given by $W_i = TF(t_i) \times IDF(t_i)$ where $TF(t_i)$ is the raw frequency of t_i in the bag of RSTP representing the utterance (i.e., 0 or 1); and $IDF(t_i) = log(\frac{N}{n_i})$ where N is the total number of utterances mined to produce the RSTP database, and n_i is the number of mined utterances including t_i in their representation.

4 Experimentation

This experimentation aims at comparing selection methods on the task of retrieving a response utterance in a large corpus of open-domain textual dialogues from a given input utterance. The dialogue corpus consists of two main types of utterances: (i) *initiative* utterances that have at least one follow-up utterance ; and (ii) *response* utterances that do not have a follow-up utterance. The retrieving process works as follows. Initiative utterances from the corpus are ranked against a given input utterance. Then, a random response is taken from the pool of response utterances of the highest ranked initiative utterance (in this work, note that 91% of the response pools are of size 1).

Evaluation aims at assessing (i) the ability of each selection method to find an initiative utterance that is close to the given input utterance, and (ii) the ability of each method to select an appropriate response to a given input utterance. This experiment compares a RSTP-based method with four other selection methods (described in Section 4.3) on a set of 1000 reference utterances. Reference utterances are the input utterances of the selection methods. Each reference utterance comes along with a (possibly large) predefined set of acceptable responses (detailed in Section 4.1). Notably, reference utterances do not appear in the selection corpus, that is, there is no initiative utterances that is strictly equal to any of the reference utterances.

For each method, assessment consists in comparing the selected response produced for a reference utterance against the list of acceptable responses associated with this reference utterance. The more similar the selected response is to *one of* the predefined acceptable responses, the better it is. To avoid a time-consuming, costly and possibly noisy human intervention at this step, we consider metrics coming from the machine translation domain such as BLEU [13] or TER [16]. The main idea behind these metrics is to measure the correspondence between a system output translation and a set of reference translations while maintaining an adequate correlation with human judgements of quality. The TER ("Translation Error Rate") metric is the most appropriate to the need of this experimentation since it targets cases where a large space of possible correct translations exists. In particular, it is not required for a selected response to be close to all the predefined acceptable responses but only to one of those. TER is defined as "the minimum number of edits needed to change a hypothesis so that it exactly matches one of the references, normalised by the average length of the references". Edits include insertion, deletion, substitution of single words and shifts of word sequences. For a given hypothesis utterance, it is given by the formula: $\text{TER} = \frac{\#\text{ of edits}}{\text{average }\#\text{ of reference words}}$.

4.1 Selection Corpus and Reference Utterances

A subset of the English version of the OpenSubtitles2016 corpus [9] was used as the selection corpus. This corpus consists of a wide variety of subtitles of television dramas. It provides a large amount of pre-processed transcribed interactions that can be useful for dialogue modelling. Pre-processing includes subtitle encoding conversion, sentence segmentation, sentence tokenisation and corrections of spelling errors [9]. Subtitles are formatted as sequences of tokenised sentences with timing information and meta-data about the subtitle (e.g., identifiant of the TV episode). In this work, pre-processing was extended by applying a named entity (NE) recognition for each sentence. This was done with the Stanford NER [4]. NEs allows to generalise sentences by replacing person name, localisation and organisation (e.g., "My name is Alice ." is turned into "My name is <person> ."). Thus, NEs stay neutral for the similarity calculations undertaken while ranking initiative utterances. However, the turn structure is missing from these subtitles which renders the OpenSubtitles2016 corpus noisy for dialogue modelling. To overcome this problem, a process similar to the one used to build the SubTle

Table 1. Figures about the selection corpus (subset of OpenSubtitles2016 [9]) and about the dataset of reference utterances. T/U=Tokens per Utterance

	Selection corpus		Reference utterances
	Initiative utterances	Response utterances	
Unique utterance	3,174,606	2,481,369	1000
Tokens (unique)	23,148,094 (226,462)	19,557,246 (219,374)	5571 (348)
T/U: avg/median (std)	7.29/7.0 (6.58)	7.88/7.0 (5.67)	5.57/5.0 (0.78)
T/U: min/max	1/1431	1/1280	5/10

corpus [1] was carried out. It aims at extracting utterance pairs corresponding to an initiative and a response exchanged in a dyadic conversation. This heuristic helped to reduce the level of noise to approximately 25% of the conversational pairs on the SubTle corpus [1]. It is based on timing features about consecutive sentences, punctuation features (such as a sentence-initial dash) and the fact that sentences are shown on the same subtitle block (i.e., appearing on the same screen). This method allows to extract exchanges of utterances that are less noisy than the entire corpus. Table 1 presents some figures about the selection corpus. It includes more than 3 million of unique initiative utterances and around 2.4 million of unique response utterances.

The set of reference utterances along with their predefined set of acceptable responses has been automatically extracted from the subset of the OpenSubtitles2016 corpus. To this purpose, we extracted the 1000 most frequent utterances from the corpus which contains at least 5 tokens (inclusive). The high frequency of these utterances ensures that they are very likely to be used in a conversation by a human. The 5-token requirement follows recent observations showing that it is difficult for a human to reliably judge the validity of a conversational pair if the first part is too small [3]. Importantly, all the retained reference utterances have been discarded from the selection corpus. Table 1 presents some figures about the reference utterances. In average, a reference utterance has 191.11 acceptable responses (std=426.94, median=102, min=62, max=7677).

4.2 The RSTP-based Method

The RSTP-based model was prepared by mining patterns on the set of initiative utterances of the selection corpus. Table 2 presents some figures about the RSTP database. First, the number of RSTP extracted from the corpus is less than 2 times the number of unique utterances. Indeed, the full database contains around 5.7 million unique patterns which amounts to 1.82 times the number of initiative utterances. If we only consider representative RSTPs that have been used to represent the initiative utterances of the selection corpus, it comes down to approx. 3.8 million of unique patterns (1.21 times the number of initiative utterances). In comparison, the number of unique trigrams extracted from the initiative utterances of the selection corpus is around 5.7 million items.

Besides, the representation of utterance with the RSTP-based model is sparse. Indeed, the number of patterns per utterance representation is in average 3.09 (std=3.24, median=3.0, min=1, max=582).

Figure 1 takes a closer look at the distribution of the size (in tokens) of RSTP used to represent initiative utterances from the selection corpus. It shows that the RSTP-based model effectively uses a wide variety of patterns in terms of size, contrary to a fixed n-gram model. Sizes of the patterns mostly range from 1 token to 8 tokens, with 50% of the patterns having a size between 3 and 5 tokens (median=4 tokens).

Table 2. Figures about the RSTP database mined on the initiative utterances of the selection corpus and on the RSTP effectively used to represent the initiative utterances.

RSTP database	Full	Used
Size	5,776,901	3,846,956
Tokens per RSTP		
... avg/median	4.77/4.0	4.57/4.0
... std, min/max	2.23, 1/157	1.96, 1/157

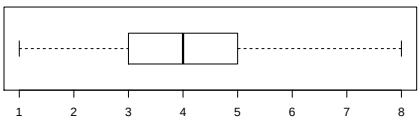

Fig. 1. Distribution of the size of the RSTP effectively used to represent the initiative utterances (in tokens, including begin and end markers). For readability, outliers have been discarded.

4.3 Other Selection Methods

Four other selection methods are considered in this experimentation. These methods differ in their way to rank initiative utterances given a reference utterance. However, they follow the same process to pick the response utterance. First, the *random method* selects a random initiative utterances from the selection corpus following a uniform distribution. Thus, it does not take into account the reference utterance given as input. Secondly, the *TF-IDF method* implements a VSM at the token level (i.e. it considers unigram). It retrieves initiative utterances that are lexically close to the reference utterance but does not take into account word order. An utterance is represented by a TF-IDF weighted vector of the unigrams that occurred in it. Let W_i be the weight assigned to unigram u_i. It is given by $W_i = TF(u_i) \times IDF(u_i)$ where $TF(u_i)$ is the raw frequency of unigram u_i in the utterance; and $IDF(u_i) = log(\frac{N}{n_i})$ where N is the total number of initiative utterances, and n_i is the number of initiative utterances containing u_i. Similarity between two utterances is given by the cosine similarity of their vector representations. Then, the *trigram method* implements a VSM at the n-gram level with n=3. It is equivalent to the previous model with the exception that it considers trigram instead of unigram and that begin and end markers are added to the utterance. This method takes into account lexical proximity between utterances and word order to the extent of trigrams. Finally, the last method relies on word and utterance embeddings using the *doc2vec model* [7]. Word and utterance embeddings are jointly learnt as the coefficients of a shallow neural network trained to predict a word given its context and the

utterance it belongs to. We focused especially in harvesting the utterance embeddings as their cosine similarity can translate lexical and semantic similarity. The implementation provided by Gensim [14] is used with the length of the context window set to 2 and a vector dimension of 100. The model was trained on the entire selection corpus. Embeddings of the reference utterances are inferred and used to retrieve the closest initiative utterance with a nearest neighbour search.

5 Results

5.1 Ranking of Initiatives and Selection of Responses

We compare the results of the ranking process operated by each selection method. This process consists in finding an initiative utterance from the selection corpus that is close to a given reference utterance. For instance, for the reference utterance "what is this about ?", the following initiatives were retrieved from the database: "– it looks like <person> !" (random), "– what about this ?" (TF-IDF), "– <person> , what is this about ?" (trigram), "– i don 't . what is this about ?" (doc2vec), and "– and what is this about ?" (RSTP). For the reference utterance "good to see you .", the following results were retrieved: "– i 'm not gonna do it this time ." (random), "– good of you to see me ." (TF-IDF), "– good to see you . thank you ." (trigram), "– good good ." (doc2vec), and "– good to see you . pleasure ." (RSTP). It should be noted that in the vast majority of the cases, the ranking processes of the methods yielded a clear-cut initiative utterance matching the reference utterance. In some marginal cases, the TF-IDF, trigram and RSTP methods yielded more than one maximum result (at most 4 for the TF-IDF model, 2 for the others). In these cases, the result of the ranking process was a random choice between those maximum results. Table 3 (columns "I") takes a closer look at the common results between methods in the ranking process. Comparison consists in strict string equality. It turns out that the ranking step of the methods lead to different results. Methods share less than 10% of their ranking results, with the exception of the TF-IDF and trigram methods that share around 17% of their results. In particular, the random method does not share results with the other ones. It shows that each method has inherent characteristics making it more or less suited for utterance selection.

Table 3. Common results between methods in the ranking of the initiative utterance (I) and in the selection of the response utterance (R). Presented results are symmetric.

	Random		TF-IDF		Trigam		doc2vec		RSTP	
	I	R	I	R	I	R	I	R	I	R
Random	–	–	0%	0%	0%	0%	0%	0%	0%	0%
TF-IDF	0%	0%	–	–	17%	17.6%	5%	5%	8.3%	8.5%
Trigram	0%	0%	17%	17.6%	–	–	3%	3%	8%	8%
doc2vec	0%	0%	5%	5%	3%	3%	–	–	4%	4%
RSTP	0%	0%	8.3%	8.5%	8%	8%	4%	4%	–	–

We now consider the impact of the methods on the quality of the selected response utterance. The selection process is the global procedure by which each method selects a response to a given reference utterance. For instance, the following responses were retrieved from the database for the reference utterance "you 're not serious .": "– no ." (random), "– i 'm serious ." (TF-IDF), "– listen to me . i am very serious ." (trigram), "– i am ." (doc2vec), and "– sorry to burst your bubble ." (RSTP). However, results may be noisier. For example, the following results were retrieved for the reference utterance "can I help you ?": "– a had accomplices ." (random), "– we 'll get her anyway ." (TF-IDF), "– we 'll get her anyway ." (trigram), "– what are you doing ?" (doc2vec), and "– yeah ." (RSTP). Table 3 (columns "R") presents the common results between methods in terms of response selection. Methods select less than 10% of the same responses except for the TF-IDF and trigram methods that share 17.6% of their responses (consistently with their ranking results). Thus, methods mostly select different responses. Table 4 gives describing figures about the datasets of selected responses by each method. Sets of selected responses by the TF-IDF, trigram, doc2vec and RSTP methods are similar. They include between 85% (doc2vec method) and 89% (trigram, RSTP methods) of unique utterances. Responses contain around 5 tokens with a minimum of 1 token and a maximum between 35 (TF-IDF method) and 47 (RSTP method) tokens. Responses selected by the random method have a more variable size in terms of tokens per utterance as shown by a higher standard deviation and by a maximum size of 101 tokens.

Finally, we consider the quality of the response selected by each method. To avoid a time-consuming and labour-intensive human evaluation, we decided to assess the quality of a selected response by comparing it to the list of acceptable responses associated with each reference utterance. To this purpose, we chose to compute for each method the "Translation Error Rate" between a selected response to a reference utterance and the list of acceptable responses. This indicator computes the minimum number of edits needed to change a selected response so that it exactly matches one of the acceptable responses. Results are presented in Table 4. TER results range from 0.505 to 0.632. The worst TER is for the random method (0.632). The best rate is for the RSTP method (0.505). TF-IDF, trigram and doc2vec methods share comparable results (between 0.53 and 0.57). We performed a paired Wilcoxon test to check for statistically significant differences between methods. TER score for the RSTP method is significantly lower than the scores from the random ($p < 0.001$), trigram ($p < 0.05$) and doc2vec ($p < 0.01$) methods. However, it is not significantly lower than the score from the TF-IDF method. TER score for the random method is significantly higher than the scores from all the other methods. All other differences are not statistically significant at the 5 percent level.

5.2 Discussion

This experimentation has aimed at comparing four selection methods (a random one, two VSM based on unigram and trigram, a GVSM on RSTP and a word embeddings model) on a task of utterance selection in a large database

Table 4. Figures about the datasets of selected responses for each method and their associated "Translation Error Rate" (TER). T/U=Tokens per Utterance

	Random	TF-IDF	Trigram	doc2vec	RSTP
Utterances (unique)	1000 (87%)	1000 (87%)	1000 (89%)	1000 (85%)	1000 (89%)
Tokens (unique)	5710 (1154)	5591 (1009)	5808 (1018)	5698 (1023)	5438 (1028)
T/U: avg/median	5.71/5.0	5.59/5.0	5.81/5.0	5.70/5.0	5.44/5.0
T/U: **std**, min/max	5.36, 1/101	3.51, 1/35	3.70, 1/42	3.77, 1/46	3.37, 1/47
TER	0.632	0.537	0.549	0.566	0.505

of open-domain dialogue pairs. Results show that these methods are inherently different in the sense that they (i) mostly retrieve different initiative utterances given a reference one, and (ii) select different response utterances. Besides, we have measured the quality of utterances selected by each method in terms of the translation error rate (TER). This indicates that the RSTP-based method is a promising approach for utterance selection. However, these results should be taken with caution. First, the acceptability of an utterance is not entirely indicated by the TER score since it ignores the notion of semantic equivalence. Assessing the acceptability of each utterance would require a more costly evaluation based on human judges. Then, even though the TER score has allowed us to clearly distinguish the random model from the other ones, the error rates obtained by non-random methods are still high. We cannot exclude the possibility that the methods have selected valid responses that were not appearing in the list of acceptable ones (thus, increasing the error rate). Indeed, open-domain utterances may accept a huge space of possible responses that may be roughly estimated by our lists of acceptable responses. On the other hand, the database of dialogue example may still be noisy to a large extent despite our effort to reduce it. However, all selection methods are equally affected by this problem. Last but not least, this experimentation compares selection method on the basis of highly frequent reference utterances. An interesting extension of this work would consider the case of less frequent utterances. Nevertheless, this would require a database of those utterances along with their acceptable responses.

6 Conclusion and Future Work

This paper has presented a new corpus-based process that aims at finding and exploiting recurrent surface text patterns of language use to represent open-domain dialogue utterances for a retrieval task. Our approach provides the benefit of being corpus-based, unsupervised, parameterless, language-independent while exploiting patterns that are easily understandable from a human perspective. We have shown that this approach performs comparatively well to other retrieval models on a task of selection of dialogue examples derived from a large corpus of written dialogues. Future work includes the study of this approach on other corpora and other languages as well as the potential of our model to more generally model dialogue history involving several utterances.

References

1. Ameixa, D., Coheur, L., Redol, R.A.: From subtitles to human interactions: introducing the subtle corpus. Tech. rep., INESC-ID (2013)
2. Banchs, R.E., Li, H.: IRIS: a chat-oriented dialogue system based on the vector space model. In: Proceedings of the ACL 2012 Demonstrations. pp. 37–42 (2012)
3. Charras, F., Dubuisson Duplessis, G., Letard, V., Ligozat, A.L., Rosset, S.: Comparing system-response retrieval models for open-domain and casual conversational agent. In: Workshop on Chatbots and Conversational Agent Technologies (2016)
4. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 363–370 (2005)
5. Gandhe, S., Traum, D.R.: Surface text based dialogue models for virtual humans. In: Proceedings of the SIGDIAL (2013)
6. Gusfield, D.: Algorithms on Strings, Trees and Sequences. Cambridge University Press, Cambridge, UK (1997)
7. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML. vol. 14, pp. 1188–1196 (2014)
8. Lee, C., Jung, S., Kim, S., Lee, G.G.: Example-based dialog modeling for practical multi-domain dialog system. Speech Communication 51(5), 466–484 (2009)
9. Lison, P., Tiedemann, J.: OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In: 10th edition of the Language Resources and Evaluation Conference (LREC). Portorož, Slovenia (May 2016)
10. Lowe, R., Pow, N., Serban, I.V., Pineau, J.: The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: SIGDIAL. p. 285 (2015)
11. Mooney, C.H., Roddick, J.F.: Sequential pattern mining – approaches and algorithms. ACM Computing Surveys 45(2), 19:1–19:39 (2013)
12. Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, a.A., Glass, J., Randeree, B.: Semeval-2016 task 3: Community question answering. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 525–545 (2016)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. pp. 311–318 (2002)
14. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
15. Salton, G., McGill, M.J.: Introduction to modern information retrieval (1986)
16. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of association for machine translation in the Americas. vol. 200 (2006)
17. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. CoRR abs/1506.06714 (2015)
18. Weizenbaum, J.: ELIZA - a computer program for the study of natural language communication between man and machine. Communications of the ACM 9(1), 36–45 (Jan 1966)
19. Wong, S.K.M., Ziarko, W., Raghavan, V.V., Wong, P.: On modeling of information retrieval concepts in vector spaces. ACM Transactions on Database Systems 12(2), 299–321 (1987)