Semantic Diversity for Natural Language Understanding Evaluation in Dialog Systems

Enrico Palumbo
Amazon Alexa
Turin, 10126, Italy
palumboe@amazon.com

Andrea Mezzalira
Amazon Alexa
Turin, 10126, Italy
mezzalir@amazon.com

Cristina Marco
Amazon Alexa
Turin, 10126, Italy
marcocri@amazon.com

Alessandro Manzotti
Amazon Alexa
Turin, 10126, Italy
manzotti@amazon.com

Daniele Amberti
Amazon Alexa
Turin, 10126, Italy
amberti@amazon.com

Abstract

The quality of Natural Language Understanding (NLU) models is typically evaluated using aggregated metrics on a large number of utterances. In a dialog system, though, the manual analysis of failures on specific utterances is a time-consuming and yet critical endeavor to guarantee a high-quality customer experience. A crucial question for this analysis is how to create a test set of utterances that covers a diversity of possible customer requests. In this paper, we introduce the task of generating a test set with high semantic diversity for NLU evaluation in dialog systems and we describe an approach to address it. The approach starts by extracting high-traffic utterance patterns. Then, for each pattern, it achieves high diversity selecting utterances from different regions of the utterance embedding space. We compare three selection strategies based on clustering of utterances in the embedding space, on solving the maximum distance optimization problem and on simple heuristics such as random uniform sampling and popularity. The evaluation shows that the highest semantic and lexicon diversity is obtained by a greedy maximum sum of distance solver in a comparable runtime with the clustering and the heuristics approaches.

1 Background

In the past years, voice-first dialog systems have become ubiquitous in the market, with an ever increasing number of features, languages and customer requests. A crucial component of these systems is the Natural Language Understanding (NLU) model. The NLU model maps customer requests onto specific actions that the device has to perform. In practice, this means classifying an utterance into a domain, intent and slots (Su et al., 2018). For instance, given the customer's utterance "play madonna", an NLU model returns: (*Music, PlayMusicIntent, play ArtistName*) where *Music* is the domain, *PlayMusicIntent* is the intent and the slot is *ArtistName*. When a new algorithm for NLU is proposed in a research environment, the evaluation is typically performed by aggregating metrics such as Slot Error Rate (SER) (Makhoul et al., 1999) and Semantic Error Rate (SemER) (Su et al., 2018) on a large test set of utterances. However, in a production environment, aggregated metrics alone are not sufficient, as they may hide failures on specific business critical utterances. Thus, whenever a change is introduced into an NLU model, failures need to be manually reviewed to determine whether they represent an issue for the customers. The manual review of failures is a crucial, and yet very time-consuming operation. Hence, the question: how to create a test set that makes the analysis more efficient including a diversity of patterns, utterances and possibile failure causes? The problem of maximizing semantic diversity in text is common in tasks such as text summarization (Zhu et al., 2007), text generation (Xu et al., 2018), keyphrase extraction (Bennani-Smires et al., 2018), machine translation (Shu et al., 2019), data augmentation in dialog systems (Hou et al., 2018; Cho et al., 2019). However, to the best our knowledge, semantic diversity has never been used to

Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pages 44–49
Barcelona, Spain (Online), December 12, 2020

create test sets for the evaluation of natural language understanding models in dialog systems.

In this paper, we introduce an approach to automate the creation of test sets with high semantic diversity for the evaluation of the NLU model in a dialog system. The approach works as follows. First, we filter the dataset extracting a set of high-traffic pattern. Then, for each pattern, we map utterances into an embedding space to represent the semantics of the different slot values. Finally, we create test sets comparing three selection algorithms based on partitioning the space in groups and selecting representatives or on directly solving a maximum sum of distance optimization problem to achieve high diversity.

2 Approach

The approach can be divided in three major steps: pattern extraction, encoding and selection (Fig. 1).

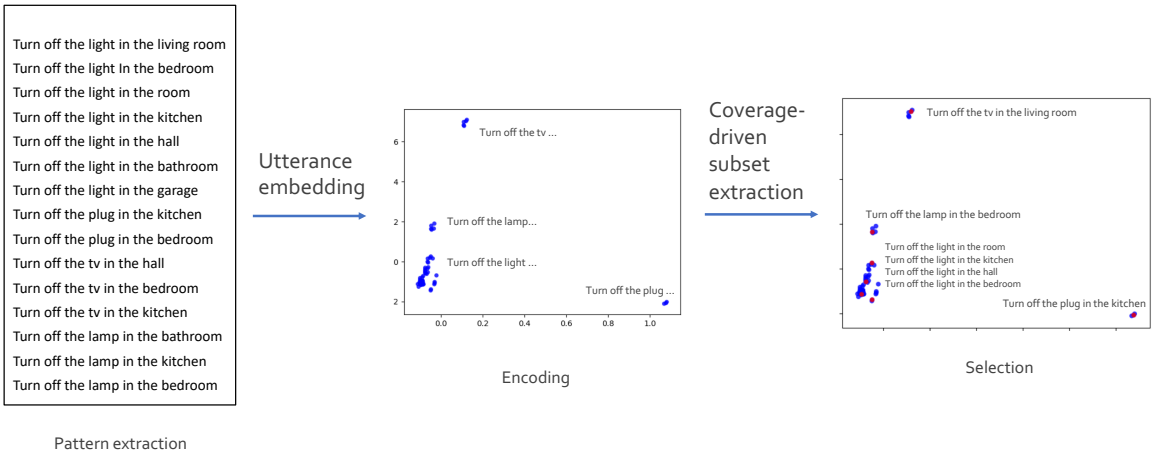

Figure 1: A bird's eye view over the proposed approach. High-traffic patterns are extracted and, for a specific pattern, utterances are embedded into a vector space in the encoding stage. Then, the selection stage selects points that are far apart in the vector space to create a test set with high diversity (red points),

2.1 Pattern Extraction

As of today, pattern-based rules such as Finite State Transducers (FSTs) (Karttunen, 2000) still play a very important role in NLU models. FSTs work by mapping into domain, intent and slots utterances that exactly match structures such as "play SongName", "play SongName please", "please can you play SongName". We call these structures "semantic frames", and, together with domain and intent, they are a suitable definition of "pattern" that can break in an FST. Given a domain $d \in D$, an intent $i \in I$ and a semantic frames $c \in C$, we define a pattern $p \in P$ as:

$$p = (d, i, c) \tag{1}$$

such as "Music, PlayMusicIntent, play SongName" or "Weather, GetWeatherForecastIntent, what is the weather like in CityName". We use a dataset composed by ~5M annotated utterances that contains ~400 high-traffic patterns. Even within a specific pattern, though, the variability can be high and the selection strategy should be diversity-aware. Consider the example of "play SongName": a huge amount of possible songs are present in the dataset. The resulting test set should include a diversity of songs, both in terms of lexicon, that is different wordings, and also in terms of semantics, for instance different musical genres.

2.2 Encoding

A crucial point for measuring diversity is finding an adequate vector representation of words and utterances where similarity metrics can be easily applied. word2vec embeddings (Mikolov et al., 2013) have

shown the effectiveness of the Continuous Bag of Words and Skip-gram architectures to learn word representations, gaining tremendous popularity. FastText (Bojanowski et al., 2017) improves the word2vec model including subword information (character n-grams) into the skip-gram architecture. In this work, we use FastText to map utterances into embeddings. This means that the model is trained to predict, given a character n-gram as input, the surrounding character n-grams in a predefined window. Given an utterance $s(p)$ of a pattern p and its K character n-grams $k_i(s(p))$ we obtain the vector representation of the utterance $\hat{s}(p)$:

$$\hat{s}(p) = \frac{1}{K}\sum_{i=1}^{K} fasttext\_pretrained\_vector(k_i(s(p))) \quad (2)$$

Currently, popular models such as ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019) further improve word representations by considering the context or embedding the whole sentence based on neighboring sentences (Kiros et al., 2015). We choose FastText over more sophisticated embedding models because it is frugal (fast at retrieval times on CPU), and it provides pre-trained models for 157 different languages. The major drawback of averaging character n-grams embeddings in this way is that we lose information on how the sentence is structured, e.g. the order of the tokens. However, given that we perform the encoding in pattern-wise manner, the structure of the sentence is fixed as described in Sec.2.1 and the variations mostly come from the values that occur in the slots.

2.3 Selection

Definition 1 *Given a pattern p, $M = |p|$ is the total number of utterances in the pattern*

Definition 2 *Given a pattern p, $m \leq M$ is the total number of utterances to be selected for the pattern*

Definition 3 *Given the vector representation of an utterance $\hat{s}(p)$, $d = |\hat{s}(p)|$ is the number of dimensions of the vector.*

Definition 4 *$X(p) = (\hat{s}_1(p), ..., \hat{s}_M(p))$ is the matrix that contains the vector representations of all the utterances in a pattern p*

We compare the following approaches to select points from the vector space:

PSA The Part and Select Algorithm (PSA) (Salomon et al., 2013) has two steps: first, it partitions the space grouping similar points; then it selects a diverse subset by choosing one member for each of the groups. To partition the space in m subsets, PSA makes $m-1$ divisions of a single set into two subsets. Given the minimum and maximum values of a feature $a_j = min_i(X_{ij})$ and $b_j = max_i(X_{ij})$, the diameter of a subset is defined as $A = max_j(b_j - a_j)$. The partitioning of the space works iteratively, searching among all the subsets the one that has the maximum diameter A, and splitting in half the subset along the feature j that maximizes the diameter. Then, for each of the m subset, the point that is closest to the center of the hyperretangle is selected. PSA has a runtime complexity that is $O(M * m * d)$.

KMeans KMeans (Hartigan and Wong, 1979) is arguably the most popular clustering algorithm, it works by dividing the data in a predefined number of groups minimizing the within-cluster sum of squares. For each pattern with M utterances, we apply KMeans to obtain m clusters, and then we select the nearest point to the centroid to be part of the subset. KMeans has a runtime complexity of $O(M * m * d)$.

MaxSum MaxSum (Ghosh, 1996) solves the optimization problem of finding a subset of points that have the maximum sum of distances among each other. Given that the problem is NP-hard, we use a greedy approach that iteratively selects points that maximize the objective and has a linear runtime complexity $O(M * m * d)$.

As baselines, we also include the **Random Sampler**, which selects m points per pattern using a uniform distribution, and the **Popularity Sampler**, which selects the most frequently used m utterances for each pattern. For all selection algorithms, we set as the default percentage of utterances to select for each pattern $f = 0.01$. Given the number of utterances in a pattern M and f, we determine the number of points to select and set the number of clusters $m = Mf$ in the clustering algorithms. When using the proposed approach, we recommend to set the value of f depending on the desired size of the test set.

3 Evaluation

We evaluate the inherent diversity of the test sets that the selection algorithms generate measuring how 'distant' two utterances are on average in the subsets that we generate using the following metrics:

- **SelfBLEU** (Zhu et al., 2018) was recently introduced to measure the diversity of artifically generated text, it computes BLEU (Papineni et al., 2002) comparing a set of utterances with themselves rather than with a reference. We use it as follows:

$$SelfBLEU = \frac{1}{N}\sum_{p=1}^{N} avg_{i,j}(1 - BLEU(s(p)_i, s(p)_j)) \tag{3}$$

- **Jaccard**: average word overlap across test utterances

$$Jaccard = \frac{1}{N}\sum_{p=1}^{N} avg_{i,j}(1 - word\_overlap(s(p)_i, s(p)_j)) \tag{4}$$

- **Word Embedding Diversity (WED)**: similar to the Word Embedding Similarity (Agirre et al., 2016), it is the average cosine distance between embeddings of vectors in the test set:

$$WED = \frac{1}{N}\sum_{p=1}^{N} avg_{i,j}(1 - cosine\_similarity(\hat{s}(p)_i, \hat{s}(p)_j)) \tag{5}$$

Note that SelfBLEU and Jaccard only consider word and n-gram level similarities, whereas WED can also take into account word semantics.

4 Results

We compare the selection algorithms computing the relative percentage improvement with respect to random selection on the diversity metrics (Tab. 1). The results show that, in general, all diversity-aware algorithms achieve higher diversity with respect to Random and Popularity generates the lowest diversity. MaxSum solver obtains the best diversity both at the semantic (WED) and at the lexicon level (SelfBLEU, Jaccard). Interestingly, PSA performs better than KMeans for metrics that take into account words and n-grams overlaps, i.e. at a lexicon level, whereas KMeans works better for WED, which measures embedding distance at a semantic level. Random is the fastest algorithm, but the runtime is comparable for all the algorithms.

Algorithm	**WED**	**SelfBLEU-2**	**SelfBLEU-3**	**SelfBLEU-4**	**Jaccard**	**Runtime (s)**
PSA	24.73	2.33	1.97	1.53	1.93	12178
KMeans	26.13	2.20	1.96	1.59	1.87	13679
MaxSum	**38.7**	**6.96**	**6.13**	**4.9**	**5.31**	12219
Random	0.0	0.0	0.0	0.0	0.0	**11959**
Popularity	-12.98	-10.88	-9.66	-7.75	-7.75	12011

Table 1: Diversity comparison of the selectiong algorithms as a relative % change with respect to Random sampling. In SelfBLEU-n, n is the size of the n-gram used. Results are significant for all pairs of algorithms and for all metrics with a paired t-test with p<0.05.

5 Conclusions

In this paper, we have introduced the problem of creating a test set with high semantic diversity to evaluate the NLU model of a dialog system. We have described the problem motivation and we have introduced an approach to address it. The experimental comparison among different diversity-aware selection algorithms

shows that the MaxSum sampler obtains the best diversity, both at the semantic (WED) and at the lexicon level (SelfBLEU, Jaccard). For all the diversity-aware approaches (PSA, KMeans, MaxSum), runtime is comparable to simple heuristics such as random and popularity selection. As a future work, we will create a ground truth to see how well our diversity metrics correlate with human judgement. The ground truth will also be key to exploring the effectiveness of hybrid approaches that combine diverisity and coverage, taking into the frequency of customer requests. We also plan to experiment with more encoding algorithms, such as frugal light-weight transformer-based approaches that have been recently proposed (Sanh et al., 2019) and have shown to better represent complex utterances.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium, October. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Eunah Cho, He Xie, John P Lalor, Varun Kumar, and William M Campbell. 2019. Efficient semi-supervised learning for natural language understanding by optimizing diversity. *arXiv preprint arXiv:1910.04196*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jay B Ghosh. 1996. Computational aspects of the maximum diversity problem. *Operations research letters*, 19(4):175–181.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. *arXiv preprint arXiv:1807.01554*.

Lauri Karttunen. 2000. Applications of finite-state transducers in natural language processing. In *International Conference on Implementation and Application of Automata*, pages 34–46. Springer.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel, et al. 1999. Performance measures for information extraction. In *Proceedings of DARPA broadcast news workshop*, pages 249–252. Herndon, VA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Shaul Salomon, Gideon Avigad, Alex Goldvard, and Oliver Schütze. 2013. Psa – a new scalable space partition based selection algorithm for moeas. In Oliver Schütze, Carlos A. Coello Coello, Alexandru-Adrian Tantar, Emilia Tantar, Pascal Bouvry, Pierre Del Moral, and Pierrick Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, pages 137–151, Berlin, Heidelberg. Springer Berlin Heidelberg.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827.

Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–676. IEEE.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949.

Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. Improving diversity in ranking using absorbing random walks. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 97–104.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.