Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Submodularity-Inspired Data Selection for Goal-Oriented Chatbot Training Based on Sentence Embeddings

Mladen Dimovski[1], **Claudiu Musat**[2], **Vladimir Ilievski**[1], **Andreea Hossmann**[2], **Michael Baeriswyl**[2]
[1] School of Computer and Communication Sciences, EPFL, Switzerland
[2] Artificial Intelligence Group - Swisscom AG
firstname.lastname@{epfl.ch, swisscom.com}

Abstract

Spoken language understanding (SLU) systems, such as goal-oriented chatbots or personal assistants, rely on an initial natural language understanding (NLU) module to determine the intent and to extract the relevant information from the user queries they take as input. SLU systems usually help users to solve problems in relatively narrow domains and require a large amount of in-domain training data. This leads to significant data availability issues that inhibit the development of successful systems. To alleviate this problem, we propose a technique of data selection in the low-data regime that enables us to train with fewer labeled sentences, thus smaller labelling costs.

We propose a submodularity-inspired data ranking function, the *ratio-penalty marginal gain*, for selecting data points to label based only on the information extracted from the textual embedding space. We show that the distances in the embedding space are a viable source of information that can be used for data selection. Our method outperforms two known active learning techniques and enables cost-efficient training of the NLU unit. Moreover, our proposed selection technique does not need the model to be retrained in between the selection steps, making it time efficient as well.

1 Introduction

In their most useful form, goal-oriented dialogue systems need to understand the user's need in great detail. A typical way of structuring this understanding is the separation of intents and slots that can be seen as parameters of the intents. Slot filling, also known as entity extraction, is finding the relevant information in a user query that is needed for its further processing by the SLU system. For example, in a restaurant reservation scenario, given the sentence *Are there any French restaurants in downtown Toronto?* as an input, the task is to correctly output, or fill, the following slots: {*cuisine: French*} and {*location: downtown Toronto*}.

The slot filling task is usually seen as a sequence tagging problem where the goal is to tag each relevant word token with the corresponding slot name using the B–I–O (Begin, Inside, Outside) convention. The table below shows how we would correctly tag the previous example sentence.

Are	there	any	French
O	O	O	B-Cuisine
restaurants	in	downtown	Toronto
O	O	B-Location	I-Location

Most methods created for slot filling are supervised and require large amounts of labeled in-domain sentences in order to perform well. However, it is usually the case that only very little or no training data is available. Annotating new sentences is an expensive process that requires considerable human effort and, as a result, achieving good performance with as little data as possible becomes an important concern.

Our solution to the data availability problem relies on a better way of selecting the training samples to be labeled. If a limited amount of resources are available, we want to enable the user to spend them in the most efficient way. More precisely, we propose a method for *ranking* the unlabeled sentences according to their utility. We measure the latter with the help of a ranking function that satisfies the principles of submodularity, a concept known to capture the intuition present in data selection. We run experiments on three different publicly available slot filling datasets: the MIT Restaurant, the MIT Movie and the ATIS datasets [Liu *et al.*, 2013a; 2013b]. We are interested in measuring the model's performance when it is trained with only a *few dozen* labeled sentences, a situation we refer to as *low-data regime*. We compare our proposed selection method to several standard baselines, including two variants of active learning.

We identify our three main contributions:

- We show that the space of raw, unlabeled sentences contains information that we can use to choose the sentences to label.
- We create a sumbodularity-inspired ranking function for selecting the potentially most useful sentences to label.

- We apply this data selection method to the problem of slot filling and prove that the model's performance can be considerably better when the training samples are chosen in an intelligent way.

In Section 3, we provide some background and details about the novel data selection technique. In Section 4, we describe the datasets used in our experiments and the baselines that we use as a comparison reference point. Finally, in Section 5, we present and discuss the obtained results.

2 Related Work

2.1 Slot Filling

Numerous models have been proposed to tackle the slot filling task, of which a comprehensive review was written by [Mesnil *et al.*, 2015]. However, the most successful methods that have emerged are neural network architectures and, in particular, recurrent neural network schemes based on word embeddings [Kurata *et al.*, 2016; Ma and Hovy, 2016; Liu and Lane, 2016; Zhang and Wang, 2016; Zhu and Yu, 2017; Zhai *et al.*, 2017]. The number of proposed model variants in the recent literature is abundant, with architectures ranging from encoder-decoder models [Kurata *et al.*, 2016], models tying the slot filling problem with the closely related intent detection task [Zhang and Wang, 2016] and even models that use the attention mechanism [Liu and Lane, 2016], originally introduced in [Bahdanau *et al.*, 2014]. The final model that we adopted in our study is a bi-directional LSTM network that uses character-sensitive word embeddings, together with a fully-connected dense and linear CRF layer on top [Huang *et al.*, 2015; Lample *et al.*, 2016]. We provide the model's specifications and more details in the appendix.

2.2 Low-Data Regime Challenges

Typically, machine learning models work reasonably well when trained with a sufficient amount of data: for example, reported results for the popular ATIS domain benchmark go beyond 95% F1 score [Liu and Lane, 2016; Zhang and Wang, 2016]. However, performance significantly degrades when little amount of training data is available, which is a common scenario when a new domain of user queries is introduced. There are two major approaches used to handle the challenges presented by the scarcity of training data:

- The first strategy is to train a multi-task model whose purpose is to deliver better performance on the new domain by using patterns learned from other closely related domains for which sufficient training data exists [Jaech *et al.*, 2016; Hakkani-Tür *et al.*, 2016]
- The second strategy is to select the few training instances that we can afford to label. One well known approach are the active learning strategies [Fu *et al.*, 2013; Angeli *et al.*, 2014] that identify data points that are close to the separation manifold of an imperfectly trained model.

In our work, we focus on the latter scenario, as experience has shown that high-quality in-domain data is difficult to replace by using other techniques. Therefore, we assume that we can choose the sentences we want to label and the main question is how to make this choice in a way that would yield the model's best performance.

3 Data Selection

3.1 Submodularity and Rankings

Let V be a ground set of elements and $F : 2^V \to \mathbb{R}$ a function that assigns a real value to each *subset* of V. F is called *submodular* if the incremental benefit of adding an element to a set diminishes as the context in which it is considered grows. Formally, let X and Y be subsets of V, with $X \subseteq Y \subseteq V$. Then, F is submodular if for every $e \in V \setminus Y$, $F(e|Y) \leq F(e|X)$ where $F(e|A) = F(\{e\} \cup A) - F(A)$ is the benefit, or the *marginal gain* of adding the element e to the set A. The concept of submodularity captures the idea of diminishing returns that is inherently present in data selection for training machine learning models. In the case of the slot filling problem, the ground set V is the *set of all available unlabeled sentences* in a dataset and the value $F(X)$ for a set $X \subseteq V$ is a score measuring the utility of the sentences in X. Submodular functions have already been successfully used in document summarization [Lin and Bilmes, 2011; 2012], and in various tasks of data subset selection [Kirchhoff and Bilmes, 2014; Wei *et al.*, 2014; 2015]. However, to the best of our knowledge, they have not yet been studied in the slot filling context.

An important hypothesis that is made when submodular functions are used for data selection is that if X and Y are two sets of data points for which $F(X) \leq F(Y)$, then using Y for training would give an overall better performance than using X. If we have a predefined size for our training set, i.e., a maximum number of samples that we can afford to label, then we would need to find the set X that maximizes $F(X)$ with a constraint on $|X|$. Cardinality-constrained submodular function maximization is an NP-hard problem and we usually have to resort to greedy maximization techniques (see [Krause and Golovin, 2014]). If the function is monotone, then the greedy solution, in which we iteratively add the element with the largest marginal gain with respect to the already chosen ones, gives a $(1 - 1/e)$-approximation guarantee [Nemhauser *et al.*, 1978]. As a byproduct, this greedy procedure also produces a *ranking* of the $n = |V|$ sentences, i.e., outputs a ranking permutation $\pi : [n] \to [n]$ that potentially orders them according to their usefulness. Therefore, a monotone submodular function F naturally defines a selection criteria π_F that is the order in which the elements are chosen with the greedy optimization procedure. In our work, we explore different data selection criteria, without limiting ourselves to properly defined submodular functions.

3.2 Sentence Similarity

Unless we perform the selection of the sentences to label based on some intrinsic attributes such as their length, an important metric we need to define is the similarity between *a pair* of sentences. The need to introduce such a measure of similarity appeared simultaneously with the development of active learning techniques for text classification. For example, in [McCallum *et al.*, 1998], the similarity between two documents x and y, their analogue to our sentences, is measured by an exponentiated Kullback-Leibler divergence between the word occurrence empirical measures $\hat{\mathbb{P}}(W = w|x)$ and $\lambda\hat{\mathbb{P}}(W = w|y) + (1 - \lambda)\hat{\mathbb{P}}(W = w)$

where λ is a smoothing parameter. In the recent literature, however, the focus has shifted to using word or sentence embeddings as a basis for almost all tasks involving text processing. The continuous (Euclidean) space embeddings have become fundamental building blocks for almost all state-of-the-art NLP applications.

In our study, we use a recently developed technique of producing sentence embeddings, `sent2vec` [Pagliardini *et al.*, 2017]. The `sent2vec` method produces continuous sentence representations that we use to define the similarity. More precisely, for every sentence $s \in V$, the `sent2vec` algorithm outputs a 700-dimensional vector embedding $e(s) \in \mathbb{R}^{700}$ that we in turn use to define the similarity between a pair of sentences x and y as follows:

$$\mathrm{sim}(x, y) = \exp\big(-\beta \|e(x) - e(y)\|_2\big) \tag{1}$$

where

$$\beta = \left(\frac{1}{|V|(|V|-1)} \sum_{u \in V, w \in V} \|e(u) - e(w)\|_2\right)^{-1} \tag{2}$$

is a scaling constant, measuring the concentration of the cloud of points in the space. It is the inverse of the average distance between all pairs of embeddings. Note that β will in general depend on the dataset, but it is not a hyper-parameter that needs to be tuned. Finally, the exponential function condenses the similarity to be in the interval $[0, 1]$ as $\beta > 0$.

For a fixed sentence s, the most similar candidates are those whose embedding vectors are the closest to $e(s)$ in the embedding space. Tables 1 and 2 present two example sentences (shown in bold), one from the MIT Restaurant and another from the MIT Movie dataset, together with their closest neighbors. Based on the numerous examples that we analyzed, it appears that the closeness in the embedding space is in line with the human perception of sentence similarity. Therefore, selecting a particular sentence for training should largely diminish the usefulness of its closest neighbors.

A 2-dimensional t-SNE projection of the cloud of points of the MIT Movie domain, together with an isolated example cluster, is shown on Figure 1. The large black triangle in the cluster is an example sentence and the darker dots are its closest neighbors. Closeness is measured in a 700-dimensional space and distances are not exactly preserved under the 2-dimensional t-SNE projection portrayed here. The cluster shown on Figure 1 corresponds to the sentences in Table 2.

3.3 Coverage Score

Once we have defined a similarity metric between sentences, it can be used in the definition of various submodular functions. An important example is the *coverage score* function

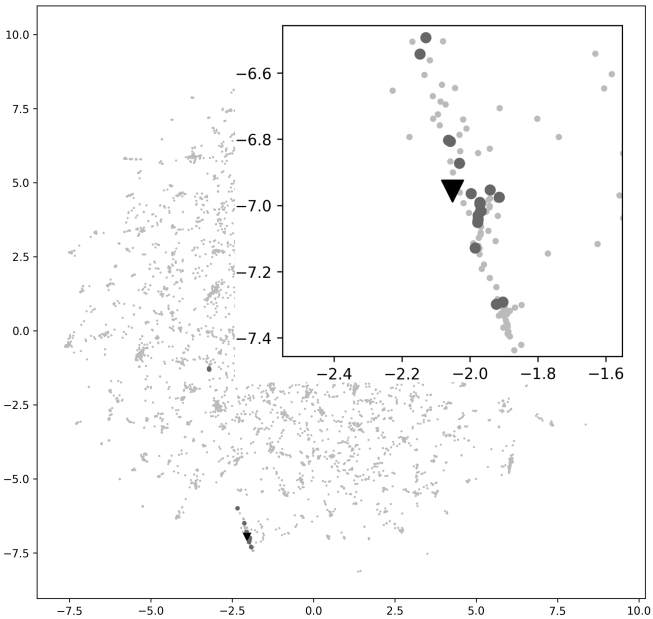

Figure 1: A 2-dimensional t-SNE projection of the cloud of points representing the embeddings of the sentences of the MIT Movie domain. The overlapping plot gives a closer view of the small cluster at the bottom left; the corresponding sentences are shown in Table 2

that evaluates every subset of sentences X in the following way [Lin and Bilmes, 2011]:

$$C(X) = \sum_{x \in X} \sum_{y \in V} \mathrm{sim}(x, y). \tag{3}$$

Intuitively, the inner sum measures the total similarity of the sentence x to the whole dataset; it is a score of how well x covers V. The marginal gain of a sentence s is given by

$$C(s|X) = C(\{s\} \cup X) - C(X) = \sum_{y \in V} \mathrm{sim}(s, y). \tag{4}$$

It can be easily seen that the marginal gain $C(s|X)$ does not depend on X, but only on s itself. This makes the function C, strictly speaking, modular. The cardinality-constrained greedy optimization procedure for C would, in this case, output the optimal solution. Table 3 presents the top three coverage score sentences from the MIT Restaurant dataset, and Figure 2 shows that these points tend to be centrally positioned in the cloud.

The coverage score function suffers from that it only considers how much a sentence is useful *in general*, and not in the context of the other sentences in the set. Hence, it might

whats a cheap mexican restaurant here
we are looking for a cheap mexican restaurant
i am looking for a cheap mexican restaurant
is ixtapa mexican restaurant considered cheap

Table 1: An example sentence s from the MIT Restaurant domain and the sentences corresponding to the three closest points to $e(s)$

what was the movie that featured over the rainbow
find me the movie with the song over the rainbow
what movie was the song somewhere out there featured in
what movie features the song hakuna matata

Table 2: An example sentence s from the MIT Movie domain and the sentences corresponding to the three closest points to $e(s)$

The top three sentences with highest coverage score
1. i need to find somewhere to eat something close by im really hungry can you see if theres any buffet style restaurants within five miles of my location
2. i am trying to find a restaurant nearby that is casual and you can sit outside to eat somewhere that has good appetizers and food that is a light lunch
3. i need to find a restaurant that serves sushi near 5 th street one that doesnt have a dress code

Table 3: The top three coverage score sentences from the MIT Restaurant domain

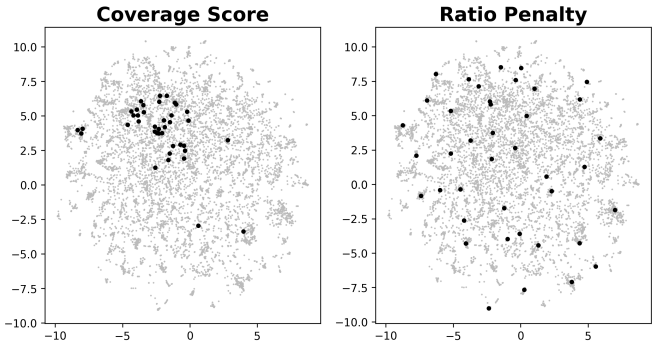

Figure 2: The position of the top forty points in the 2-dimensional t-SNE projection of the cloud corresponding to the MIT Restaurant dataset according to two different rankings

pick candidates that cover much of the space, but could be very similar to each other. In order to deal with this problem, an additional penalty or diversity reward term is usually introduced and the marginal gain takes the form of

$$D(s|X) = \sum_{y \in V} \text{sim}(s, y) - \alpha \sum_{x \in X} \text{sim}(s, x) \qquad (5)$$

where α is a parameter that measures the trade-off between the coverage of s and its similarity to the set of already chosen sentences X. The additional term makes the function D submodular as $D(s|X)$ is decreasing in X. However, D features an additional parameter that needs to be tuned and moreover, experiments with both C and D (see Figure 6 in the results section) yielded mediocre results. In the next section, we introduce our new selection method that features a non-linear penalization and does not have any additional parameters.

3.4 Ratio-Penalty Marginal Gain

We propose a direct definition of a marginal gain of an element with respect to a set. This is an alternative to providing a submodular function for which we derive the marginal gain expression and then use it to find the maximizing set of a given size. We then use this marginal gain to rank and select the sentences that are likely the most valuable for labeling. We call it the *ratio-penalty* marginal gain and define it as:

$$F(s|X) = \frac{\sum_{y \in V} \text{sim}(s, y)}{1 + \sum_{x \in X} \text{sim}(s, x)}. \qquad (6)$$

We cannot uniquely define a submodular function that generates the ratio-penalty marginal gain. To see this, notice that, for example, $F(a|\emptyset) + F(b|\{a\}) \neq F(b|\emptyset) + F(a|\{b\})$, which already makes the definition of $F(\{a, b\})$ ambiguous. Nevertheless, the above expression (6) satisfies the submodularity condition by being decreasing in X.

Domain	#train	#test	#slots	F1 score
MIT Restaurant	7661	1522	17	80.11
MIT Movie	9776	2444	25	87.86
ATIS	4978	893	127	95.51

Table 4: The three domains and their basic information (number of training samples, number of samples in the test set and number of different slots). The MIT Restaurant and MIT Movie datasets contain user queries about restaurant and movie information. The Airline Travel Information Services (ATIS) dataset mainly contains questions about flight booking and transport information

The ratio-penalty marginal gain is again a trade-off between the coverage score of a sentence and its similarity to those already chosen. An important change is that the penalty is introduced by division, instead of by subtraction as in $D(s|X)$ from the previous section. To gain more intuition, notice that, as the logarithm function is increasing, the induced ranking π_F is the same as the ranking $\pi_{\tilde{F}}$ produced by $\tilde{F}(s|X)$, where we define $\tilde{F}(s|X) = \log F(s|X)$, i.e.,

$$\tilde{F}(s|X) = \log \sum_{y \in V} \text{sim}(s, y) - \log \left(1 + \sum_{x \in X} \text{sim}(s, x)\right). \qquad (7)$$

In summary, we start with the Euclidean distance between the embeddings of the sentences, we re-scale it by dividing it by the average distance $1/\beta$ and squash it exponentially. This defines our similarity metric. Then, we do an aggregate computation in this new space by summing the similarities, and we revert to the original scale by taking the logarithm.

4 Experiments

4.1 Datasets and Method

We experiment with three different publicly available datasets whose details are shown in Table 4. Each one contains few thousand training samples, out of which we select, following some selection criteria π, only a few dozen. More precisely, we are interested in the model's performance when it is trained only with $k = 10, 20, 30, ..., 100$ labeled sentences. This selection simulates the behaviour of a system that needs to be trained for a newly available domain. We measure the performance by the best achieved F1 score during training on a separate test set that we have for each domain. The final column of Table 4 shows the performance of our adopted model trained on the *full* datasets. These are the best results that we can hope to achieve when training with a proper subset of training samples. We will denote by $\mathcal{T}_n = \{x_1, x_2, ..., x_n\}$ the full training set of a particular domain, assuming that each x_i is a (sentence, tags) pair. The training set comprising the subset of k training samples that we select using the ranking π will be denoted by $\mathcal{T}_k^\pi = \{x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(k)}\}$.

4.2 Baselines

To the best of our knowledge, data selection techniques have not yet been applied to the slot filling problem. As a result, there is no direct benchmark to which we can compare our method. Instead, we use three baselines that were used in similar situations: random data selection, classic active learning and an adaptation of a state-of-the-art selection method – randomized active learning [Angeli *et al.*, 2014].

Random Selection of Data

To select k sentences out of the total n, we uniformly pick a random permutation $\sigma \in \mathcal{S}_n$ and take as our training set $\mathcal{T}_k^\sigma = \{x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(k)}\}$. Although random data selection occasionally picks irrelevant sentences, it guarantees diversity, thus leading to a reasonable performance. We show that some complex methods either fail to outperform the random baseline or they improve upon it only by a small margin.

Classic Active Learning Framework

For our second baseline, we choose the standard active learning selection procedure. We iteratively train the model and select new data based on the model's uncertainty about new unlabeled points. To calculate the uncertainty for a sentence $x = (w_1, w_2, ..., w_k)$ containing k word tokens, we adapt the least confidence (LC) measure [Fu *et al.*, 2013; Culotta and McCallum, 2005]. We define the uncertainty of a trained model $\mathbf{\Theta}$ for the sample x as the average least confidence uncertainty across the labels

$$u(x) = \frac{1}{k}\sum_{i=1}^{k}(1 - p_{\mathbf{\Theta}}(\hat{y}_{w_i}|x)) \tag{8}$$

where $\hat{y}_w$ is the most likely tag for the word w predicted by $\mathbf{\Theta}$ and $p_{\mathbf{\Theta}}(\hat{y}_w|x)$ is its softmax-normalized score output by the dense layer of the model network.

We proceed by selecting batches of ten sentences, thus performing a *mini-batch adaptive active learning* [Wei *et al.*, 2015] as follows. The first ten sentences $\mathcal{T}_{10}$ used for the initial training of the model $\mathbf{\Theta}$ are picked randomly. Then, at each iteration $k \in [20, 30, 40, ..., 100]$, we pick a new batch of ten sentences $\mathcal{N}$ for which $\mathbf{\Theta}$ is the most uncertain and *retrain* the model with the augmented training set $\mathcal{T}_k = \mathcal{T}_{k-10} \cup \mathcal{N}$.

Randomized Active Learning

In the text processing scenario, the classic active learning algorithm has the drawback of choosing sentences that are not good representatives of the whole dataset. The model is often the least certain about samples that lie in sparsely populated regions and this blindness to the input space density often leads to a poor performance. To address this problem, data is usually selected by a weighted combination of the uncertainty about a sample and its correlation to the other samples [Fu *et al.*, 2013; Culotta and McCallum, 2005]. However, this approach requires finding a good correlation metric and also tuning a trade-off parameter of confidence versus representativeness. The latter is not applicable in our scenario because we would need to access a potential validation set, which deviates from our selection principle of labeling the least data possible. Instead, we adapt a well-performing technique proposed by [Angeli *et al.*, 2014] in which samples are selected randomly *proportionally to the model's uncertainty*. More precisely, a sample sentence x is selected with probability $u(x)/\sum_{y \in V} u(y)$. Although uncertainty will again be the highest for the poor samples, as their number is small, they will contain only a tiny percent of the total uncertainty mass across the whole dataset. Consequently, they will have very little chance of being selected.

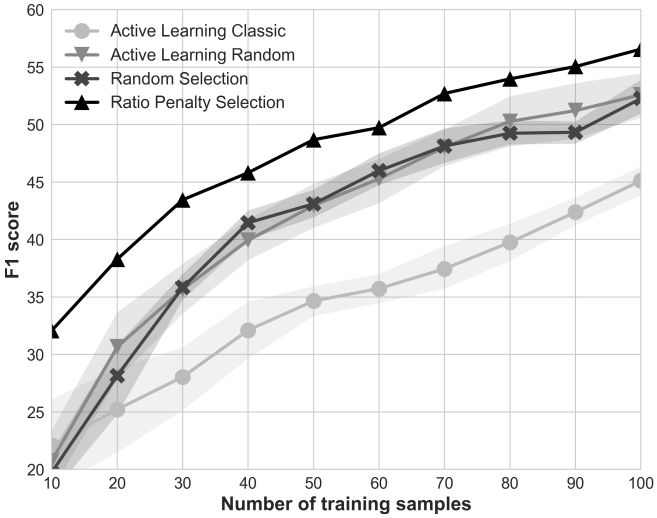

Figure 3: Comparison between the ratio-penalty and the three baseline selection methods for the MIT Restaurant dataset

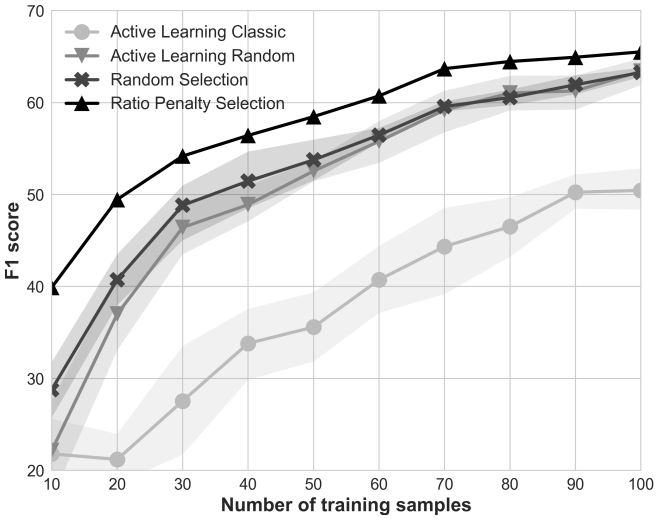

Figure 4: Comparison between the ratio-penalty and the three baseline selection methods for the MIT Movie dataset

5 Results

Figures 3, 4 and 5 show the resulting curves of the three baselines and the ratio-penalty selection (RPS) for the MIT Restaurant, the MIT Movie and the ATIS dataset, respectively. The x-axis shows the number of samples used for training and the y-axis shows the best F1 score obtained dur-

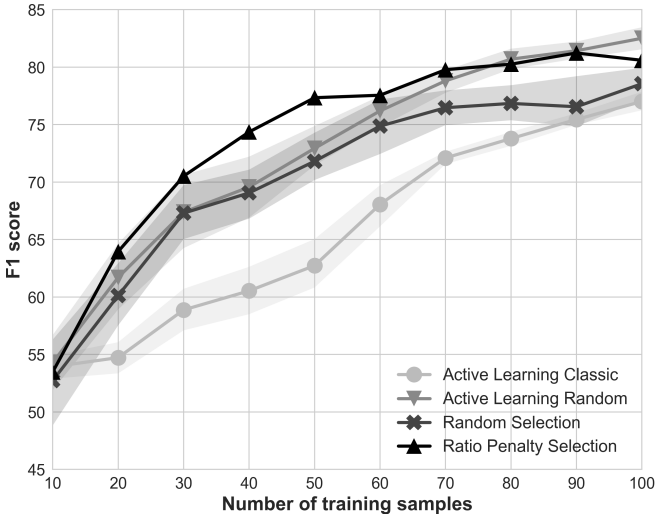

Figure 5: Comparison between the ratio-penalty and the three baseline selection methods for the ATIS dataset

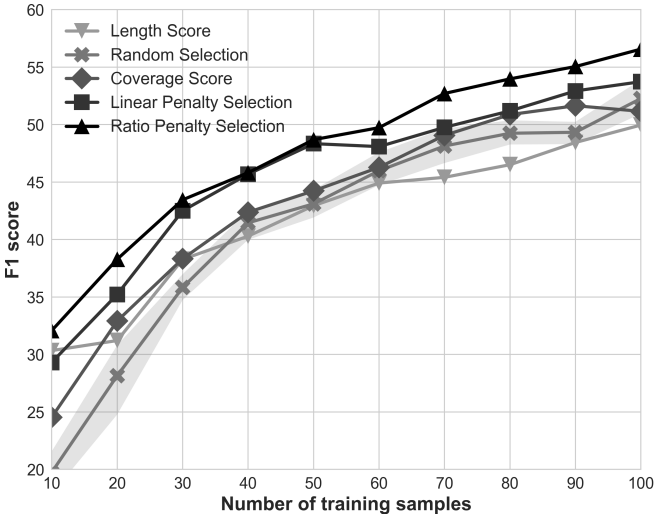

Figure 6: Comparison of various selection techniques applied on the MIT Restaurant dataset

ing training on a separate test set. As the three baselines rely on random sampling, we repeat the procedure five times and plot the mean together with a 90% confidence interval. As expected, the classic active learning (ALC) algorithm performs poorly because it tends to choose uninformative samples at the boundary of the cloud. The randomized active learning (ALR) gives a much better score, but surprisingly, it remains comparable to the random data selection strategy. Finally, the ratio-penalty selection (RPS) yields the best results, outperforming the baselines by a significant margin across all three domains. For example, in the MIT Restaurant domain, the average gap between RPS and the best performing baseline, ALR, is around 6 points in F1 score. The best result is obtained for $k = 10$ samples, where we observe approximately 55% relative improvement of RPS over ALR. Both in the MIT Restaurant and MIT Movie domains, RPS needs, on average, 20 labeled sentences *less* to match the performance of ALR.

Figure 6 presents the resulting curves of different selection strategies for the MIT Restaurant dataset. The results were similar for the remaining two domains. The *linear penalty* selection, introduced in Section 3.3 and shown only for the best parameter α, yields results that are better than the random choice and, in some regions, comparable to RPS. However, the disadvantage of this method is the requirement to tune an additional hyper-parameter that differs from one domain to another. We also present the performance when we train the model with the top k *longest* sentences (Length Score). It is fairly well in the very low data regimes, but it soon starts to degrade, becoming worse than all the other methods.

6 Conclusion

In this paper, we have explored the utility of existing data selection approaches in the scenario of slot filling. We have introduced a novel submodularity-inspired selection technique, showing that a good choice of selection criteria can have a strong influence on the model's performance in the low-data regime. Moreover, we have shown that it is not necessary to limit this search to properly defined submodular functions. As they are efficiently optimized in practice by using the derived maximal gain, defining only the latter is sufficient to produce the rankings of the sentences. In addition, we have also defined a similarity metric between pairs of sentences using their continuous vector representations produced by a recent novel technique, `sent2vec`. Finally, we have shown that the space of raw samples already contains much information that can be exploited. This information is potentially more useful than the output space information used in the active learning paradigm.

Acknowledgements

We would like to thank Patrick Thiran (EPFL), for sharing his valuable ideas and insights during the course of this research.

A Adopted Model Specifications

Our adopted model uses pre-trained GloVe embeddings on both word ($d_w = 300$) and character level ($d_c = 50$). The final word representation is a concatenation of its GloVe embedding and its character representation obtained by a separate mini-BiLSTM network. The hidden layer sizes of the main and the character BiLSTM are $h_w = 200$ and $h_c = 100$, respectively. Both the word and the character embeddings are fine-tuned during the training phase. Gradients are clipped to a maximum norm of 5 and the learning rate of the Adam optimizer, which starts at 0.005, is geometrically decayed at every epoch by a factor of 0.95, until it reaches the minimum set value of 0.001. Mini-batch size is equal to 20.

References

[Angeli *et al.*, 2014] Gabor Angeli, Sonal Gupta, Melvin Jose, Christopher D Manning, Christopher Ré, Julie Tibshirani, Jean Y Wu, Sen Wu, and Ce Zhang. Stanford's 2014 slot filling systems. *TAC KBP*, 695, 2014.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Culotta and McCallum, 2005] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

[Fu *et al.*, 2013] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and information systems*, pages 1–35, 2013.

[Hakkani-Tür *et al.*, 2016] Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *INTERSPEECH*, pages 715–719, 2016.

[Huang *et al.*, 2015] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[Jaech *et al.*, 2016] Aaron Jaech, Larry Heck, and Mari Ostendorf. Domain adaptation of recurrent neural networks for natural language understanding. *arXiv preprint arXiv:1604.00117*, 2016.

[Kirchhoff and Bilmes, 2014] Katrin Kirchhoff and Jeff Bilmes. Submodularity for data selection in statistical machine translation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141, 2014.

[Krause and Golovin, 2014] Andreas Krause and Daniel Golovin. Submodular function maximization., 2014.

[Kurata *et al.*, 2016] Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. Leveraging sentence-level information with encoder lstm for semantic slot filling. *arXiv preprint arXiv:1601.01530*, 2016.

[Lample *et al.*, 2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[Lin and Bilmes, 2011] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[Lin and Bilmes, 2012] Hui Lin and Jeff A Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012.

[Liu and Lane, 2016] Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*, 2016.

[Liu *et al.*, 2013a] Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. Asgard: A portable architecture for multilingual dialogue systems. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8386–8390. IEEE, 2013.

[Liu *et al.*, 2013b] Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. Query understanding enhanced by hierarchical parsing structures. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 72–77. IEEE, 2013.

[Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

[McCallum *et al.*, 1998] Andrew McCallum, Kamal Nigam, et al. Employing em and pool-based active learning for text classification. In *ICML*, volume 98, pages 350–358, 1998.

[Mesnil *et al.*, 2015] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539, 2015.

[Nemhauser *et al.*, 1978] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

[Pagliardini *et al.*, 2017] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*, 2017.

[Wei *et al.*, 2014] Kai Wei, Yuzong Liu, Katrin Kirchhoff, Chris Bartels, and Jeff Bilmes. Submodular subset selection for large-scale speech training data. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3311–3315. IEEE, 2014.

[Wei *et al.*, 2015] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963, 2015.

[Zhai *et al.*, 2017] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. In *AAAI*, pages 3365–3371, 2017.

[Zhang and Wang, 2016] Xiaodong Zhang and Houfeng Wang. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, pages 2993–2999, 2016.

[Zhu and Yu, 2017] Su Zhu and Kai Yu. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5675–5679. IEEE, 2017.