Artificial Intelligence 348 (2025) 104422

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Unsupervised sentence selection for creating a representative corpus in Turkish: An active learning approach

Hayri Volkan Agun

Computer Engineering Department, Bursa Technical University, Bursa, 16310, Turkey

ARTICLE INFO

Keywords:
Unsupervised active learning
Language models
Natural language processing

ABSTRACT

In this study, active learning methods adapted for sentence selection of Turkish sentences are evaluated through language learning with neural models. Turkish is an agglutinative language with a complex morphology, where the linguistic properties of words are encoded in suffixes. The active learning methods based on regression, clustering, language models, distance metrics, and neural networks are applied to unlabeled sentence selection. In this respect, a sentence corpus is selected from a larger corpus, with the same number of samples for each target word in intrinsic and extrinsic evaluation tasks. The selected sentences are used for the training of SkipGram, CBOW, and self-attention LSTM language models and extracted embeddings are evaluated by the semantic analogy, POS and sentiment analysis tasks. The evaluation scores of the models trained on the samples selected by the active learning method are compared. The results of the selected sentences based on language models indicate an improvement over random selection based on a static vocabulary. These results also show that the selection affects the quality of unsupervised word embedding extraction even if the target vocabulary is kept the same. Along with the accuracy, the time efficiency of the language models is shown to be better than other methods especially methods based on neural network models, and distance metrics.

1. Introduction

Machine learning models require annotated instances for training. The cost of annotation in supervised tasks is high and manually annotated datasets can be biased. To balance the label distributions and reduce the number of training instances, the training samples must be carefully selected. In this respect, active learning methods sample the most useful instance for labeling or annotation which will eventually improve the evaluation performance the most [1]. Most of the studies of active learning are focused on supervised learning tasks [2]. Unlike supervised machine learning tasks, unsupervised learning approaches do not require an annotation but the cost of training a representative model is generally high in terms of required sample size. Increased sample size may contain noise or repetitive dependencies and may be biased as a result pretraining may reduce the performance [3]. Adaptation of active learning methods for selection may yield better accuracy and efficiency of unsupervised learning tasks and can be used to create the most representative dataset.

An active learning approach aims to create high-quality, representation-rich, and uniform distributions [4]. However, not every active learning method is appropriate for unsupervised learning tasks. Supervised tasks which have diverse sets of labels/classes may dependent on a shallow set of hidden factors. The hidden factors through sampling can be modeled easily but sometimes the diversity

E-mail address: hayri.agun@btu.edu.tr.

https://doi.org/10.1016/j.artint.2025.104422
Received 24 May 2023; Received in revised form 10 September 2025; Accepted 17 September 2025

Available online 22 September 2025

of interactions between hidden factors is high [5]. In these cases, the learning model must be able to capture all these relations in the dataset. An example of such a model is observed in language modeling. In language modeling, the meaning of a sentence is captured through the syntactic and semantic relations between words. Based on Chomsky's hierarchy a natural language is represented by a context-sensitive grammar and is computationally complex and defined by Catalan numbers in terms of processing and combinatorial space respectively. Thus, the selection of the representative combination of words is much harder than selecting the right sample as in supervised learning.

In this study, active learning methods are compared in the selection of Turkish sentences for word representation learning via a Word2vec [6] and self-attention LSTM neural language model [7].[1] The primary goal of using active learning is to select the optimum number of sentences from a very large set that can represent the learning space the most. Since the objectives of language modeling are diverse, several active learning methods are compared. These methods are derived from their supervised counterparts and modified to work on streaming unsupervised samples. Each scoring approach in the active learning model contains different aspects such as influence of the order of words [8], having objectives of reducing the energy function, modeling based on uncertainty sampling or entropy measures [2], based on clustering [9], based on regression [10], based on distance metrics [1], and use of committee models [11]. The evaluation is done by training a neural language model on selected samples of the active learning method and comparing the learned neural language model on a separate evaluation task. The performances of the selection methods are measured independently for varying numbers of training sentences.

Active learning methods adapted with different objectives are compared according to target tasks' vocabulary, dataset size and density. The baseline approach selects the sentences randomly where as other approaches use a scoring function to select the sentences. The analogy, part-of-speech (POS) tagging and sentiment analysis tasks indicate that methods based on perplexity, KL-divergence and entropy have major improvements over the baseline selection. The improvements are also acknowledged with the self-attention neural networks in POS tagging task. The word count comparisons show the significance of the word density for each selection method and evaluation task. Additional to the accuracy scores, the average time efficiency indicates the speed-up of the KL and perplexity implementations.

The consecutive sections are dedicated to related works, the language processing steps, and active learning methods. The experimental evaluations are given in the experiments and the research findings are followed in the conclusion.

2. Related works

Active learning is one of the most studied research topics in the machine learning community. The initial research articles date back to the 1980s. Due to the recent advances in deep neural networks and machine learning methods, active learning is still a trending topic in the research community. Active learning has shared content in semi-supervised learning [4,12,13] and domain adaptation [14]. Traditional active learning methods can be categorized based on the learning methodology (i.e., supervised, semi-supervised, unsupervised), based on the processing pipeline (i.e. streamed or offline), based on the objective (i.e. inference-based, gradient-based, statistical, similarity) and based on the task (i.e., sequential learning, multi-label classification, etc.). Along with these categories, active deep learning is gaining attention as a separate research field. The research contents in each category are summarized in the following subsections.

2.1. Machine learning approach

Active learning based on machine learning types is categorized as supervised and unsupervised. The most common active learning scenario is applied through the use of the labels of a given supervised classification task. In this respect, how confident an active learning method is in the target instance label is a good indicator for the selection of this instance. This type of active selection is referred to as uncertainty sampling and has diverse implementations [2]. On the other hand, in unsupervised approaches, language models are commonly applied to score the sample. Entropy-based measures are also particularly applied in text selection [15]. Graph clustering [16], subspace learning [17], uncertainty sampling [18], gradient [19], linear representation learning [20,21], and neural network loss [22] are other types of unsupervised active learning approaches.

2.2. The type of classification

Machine learning tasks can be categorized based on assumptions about the distribution of instances, the interactions between labels, and the number of class labels per instance. Common active learning scenarios involve multi-label, multi-class [23], or single-class tasks [24]. Active learning is also widely applied in sequential classification tasks (i.e., boundary detection, named entity recognition, etc.) [4]. Another task-specific scenario, frequently observed in text classification, is sampling the most representative instances given a prior domain [15]. These approaches balance selection by considering both the representativeness and the informativeness of the queried instance.

[1] The software is provided in https://github.com/volkanagun/ActiveSelection.

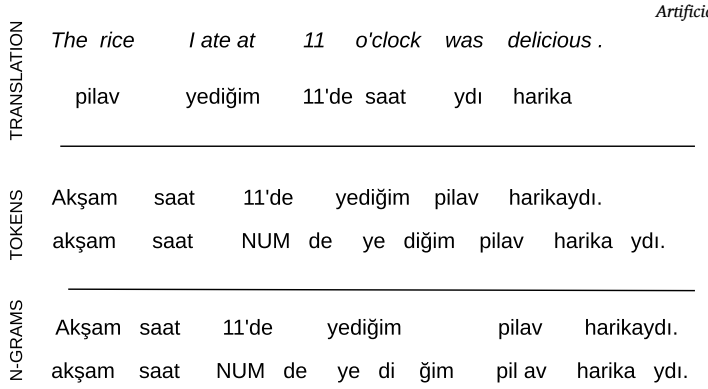

Fig. 1. An example sentence partitioning through tokenization. The sentence "The rice I ate at 11 o'clock was delicious." is tokenized by partitioning the words into lemma and suffix or into frequent n-grams.

2.3. The model objective

There are various objectives, from simple to complex, to be ensured by an active learning model. The most common objective is to reduce the uncertainty of class labels. The second is to increase the diversity of the selected samples. The third is to maximize the gradient step in supervised classification [25] or reduce the learning loss [26]. Each of these objectives is achieved with different methods. Uncertainty-based active learning uses classification margin through either decision boundary [27] or the likelihood of the model [28,29]. Other approaches in uncertainty-based active learning include prediction uncertainty of the missing values in the queried sample [30], entropy [31], and learning through generative classifiers [32]. Several approaches increase the diversity of the selected samples by using clustering [33], and specialized (committee) models [11]. Along with these objectives, deep learning approaches use data distributions or constraints (losses) over model parameters, such as KL-divergence and variational Bayesian inference objectives to weight the instances [34].

2.4. Pool or streaming data

The common use of active learning is to select the most valuable sample to annotate or label by a human expert so that the new machine learning model can generalize the most. Recent active learning approaches not only use this goal but also try to optimize the training process so that the class distributions become balanced and the data becomes better represented after the selection. To achieve these goals, active learning approaches use a pool to sample labeled instances or decide to label from streaming instances [35]. Pool-based approaches use all the details of the samples in the pool to decide on the selected samples [36]. Streaming approaches use a threshold to weigh and select the samples instantly [37,38].

3. Language processing

The processing steps start with dictionary-based tokenization. The longest dictionary entry is selected as the lemma of the token, and the rest is treated as a suffix sequence. The suffix is not removed, and it is not partitioned. An example sentence tokenization is given in Fig. 1. In tokenization, the numbers are converted to predefined symbols, and also the named entities are kept as they are without suffix partitioning. Further partitioning of the tokens into suffixes is done through filtering frequent n-grams. Frequent n-grams are given in the last example of Fig. 1.

In agglutinative languages such as in Turkish, the standard tokenization approach cannot accurately capture the morphological markers in suffixes. Morphological markers encode subcategorization, tense/aspect of the verbs and they are useful in syntactic tasks. To encode the morphological properties, the words are further partitioned into frequent n-grams. In the evaluation of active selection methods frequent n-gram partitioning is used. Frequent n-grams include suffixes such as cases, possessive, and verbal suffixes and they are also encoded by the word embedding extraction method in the given sentence context.

Due to large dictionary size, before the selection, the sentences are converted to embeddings through averaging the embeddings of the n-grams looked up from a large embedding lexicon. The averaging is done through averaging the consecutive embedding vectors within a predefined window. After the averaging is done, the averaged embedding matrix is used as a single vector in the active learning methods. For computational efficiency, embedding conversion is used in Mahalanobis distance, neural networks and least squares regression methods on the other hand the bag-of-words is used in Euclidean distance, clustering, and language models.

4. Methods

The active learning objectives based on distance metrics, language models, clustering, and neural network models are converted into a score function. Each of these conversions is explained in this section.

4.1. Distance metrics

To measure the distance between the density of the selected samples and the queried sample, Euclidean and Mahalanobis distances are used. Euclidean distance is a symmetric and positive distance measure. It is given in Equation (1).

$$distance = \sqrt{\sum_{i=1}^{n}(x_i - \mu_i)^2} \tag{1}$$

Mahalanobis distance measures the distance between the distribution of a set of samples and the queried sample. It uses the covariance matrix of the features of the sample set. It is given in Equation (2).

$$distance = \sqrt{(\vec{x} - \vec{\mu})^T S_m^{-1}(\vec{x} - \vec{\mu})} \tag{2}$$

In Equations (1) and (2), the sample is represented by the $\vec{x}$ vector, and the means or the density of the selected samples are represented by $\vec{\mu}$ vector. In Equation (2), the S^{-1} is the covariance matrix that is obtained from the selected samples. The distance measures are required to score the queried sample. If the sample is similar to the selected sample set, then it is discarded. If the distance of the queried sample is large, then it has a higher chance of selection. To obtain a proportional score, the distance measures are inverted in Equation (3).

$$score = \frac{1}{distance} \tag{3}$$

4.2. Language models

Perplexity is the main objective for selecting the most uncertain samples from the queried set. Perplexity measures how likely a target sample is according to the language model. In the proposed approach, the perplexity of the bigram language model is used. The probability of a bigram model, which is composed of the probabilities of the word pairs is given in Equation (4). The perplexity of the model is given in Equation (5).

$$P(W) = Pw_1 * P(w_2|w_1) * P(w_3|w_2)...P(w_n|w_{n-1}) \tag{4}$$

$$\begin{aligned} ln(P(W)) &= \sum_{i=1}^{n} ln(P(w_i|w_{i-1})) \\ PP(W) &= exp^{\frac{ln(P(W))}{N}} \end{aligned} \tag{5}$$

Although the perplexity measure is a robust evaluation method for language models, it is not appropriate for semantic relations between word choices, especially when there are ambiguous words. To approximate the contextual differences, several specialized models are built from the selected samples at each selection through the principal of the query by the committee. In this category, the voted entropy measure is a common approach to score a sample through possible labels obtained from different committee models. In unsupervised cases, this approach is modified based on the agreement in next-word prediction. The following modified version is given in Equation (6).

$$voted - entropy(W) = \frac{-1}{N}\sum_{i=1}^{N}\sum_{k=1}^{M} P(w_i|w_{i-1};m_k)log(P(w_i|w_{i-1};m_k)) \tag{6}$$

In the voted-entropy approach, each word probability in the sequence is summed over each model. In this case, if there is bias along the models, then the entropy will be lower. Lower entropy will drop the score of the queried instance. To measure the agreement with the different language models, another scoring method, namely Kullback-Leibler (KL) divergence, is applied. Equation (7) is given for sequence KL divergence where C represents the union model and m_k represents the current model. The agreement or disagreement between the current model and the union model can be easily computed through KL divergence.

$$KL(W) = \frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{N} P(w_i|w_{i-1};m_k)log(\frac{P(w_i|w_{i-1};m_k)}{P(w_i|w_{i-1};C)}) \tag{7}$$

When the sum of all the agreements between the union model and other models is high, the queried instance has uncertainty and it can be selected. In other cases, the queried instance may be said to be biased toward another model that has more similar patterns. When a queried instance is selected, the union model and the most similar model are updated by the sequence in the instance. Continuous updates to both the most similar model and the union model will sustain the chance of new selections.

4.3. Clustering and regression

To measure the disagreement between specialized models, we applied an online version of k-means (K-Means) clustering [39]. In the k-means clustering approach, Euclidean distance is used to measure degree of group membership. If the membership measures to

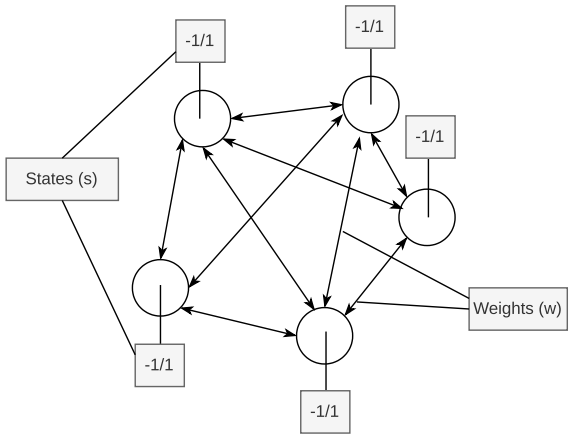

Fig. 2. An overview of a Hopfield network with five states.

the closest cluster centers are low, then the sample has a disagreement; thus it is selected, added to the closest cluster and the cluster center is updated. Through the continuous selection of the query samples, the clusters become separated from each other, and the disagreement rate will drop. The formulation of the K-Means scoring function is given in Equation (8) and Equation (9).

$$similarity(x, \mu_i) = ||\vec{\mu_i} - \vec{x}|| \tag{8}$$

$$score(x) = -\sum_{\mu_i \in means} p(x, \mu_i) * log(p(x)), where$$
$$p(x, \mu_i) = \frac{similarity(x, \mu_i)}{\sum_{\mu \in mu} similarity(x, \mu)} \tag{9}$$

Regression is one of the approaches used in active learning [40,41]. In this study, a least squares approach is used for one-class prediction. The prediction indicates the degree of membership of the sample to the selected set. If the score is large, then the sample is selected. The least squares solution for regression is given in Equation (10).

$$W = (X^T X)^{-1} X^T \vec{1}$$
$$score(\vec{x}) = \sum |W\vec{x} - \vec{1}| \tag{10}$$

4.4. Neural networks

4.4.1. Hopfield networks

Hopfield neural networks are a type of neural network also called dense associative memory networks. They are used for feature extraction and representation learning [42]. Hopfield nets consist of densely connected states and are commonly used for feature extraction. The structure of Hopfield networks is given in Fig. 2.

The Hopfield network learns through the energy function which decreases toward which gets lowered towards the local minimum. The energy function is given in Equation (11). In this equation, s_i represents the output of state i, x_i represents the input to the state i, and w_{ij} represents the connection weight between the ith and jth states.

$$Energy = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} s_i s_j - \sum_{i=1}^{n} x_i s_i \tag{11}$$

The energy function is used to score the input embedding vector of a sentence. If the energy value is low, then it is said to have a larger step in the direction of the lower minima and the input is a good candidate for storing.

4.4.2. Restricted Boltzman machine

The restricted Boltzmann machine (RBM) is a neural network model for generative learning of hidden representations in the input data. It consists of hidden and visible units, and unlike a typical feedforward neural network, the bidirectional connections between visible units and hidden units form a bipartite graph structure. An example RBM structure is given in Fig. 3.

In this paper, the energy function of a binary RBM is used as a selection criterion. The energy function of the RBM is given in Equation (12). The training of the RBM is performed through contrastive divergence with Gibbs sampling [43,44]. Each update in the contrastive divergence is given in Equation (13).

$$Energy = \frac{-1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j \tag{12}$$

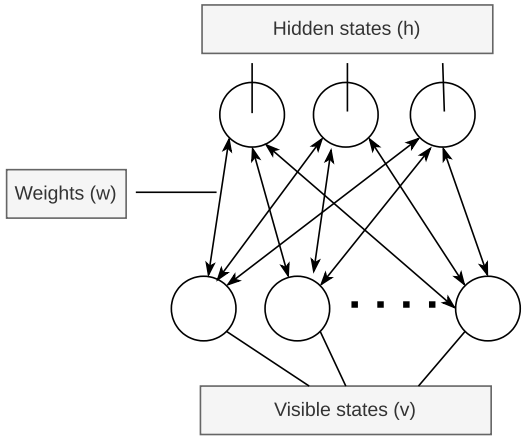

Fig. 3. An overview of a restricted Boltzmann machine with three hidden states.

$$\Delta w_{ij} = \Delta w_{ij} + \eta \cdot (sgm(\sum_{j=1}^{m} w_{ij} {v_j}^a) - sgm(\sum_{j=1}^{m} w_{ij} v_j^k)) \tag{13}$$

In Equation (13), w_{ij} is the weight between the ith hidden unit and the jth visible unit, and it is summed for every visible unit which the ith hidden unit is connected. In contrastive divergence learning the visible unit states are continuously calculated through k simulations. The state value of the visible unit at the kth simulation is given by v_j^k, and the current value is given by v_j^a. The η denotes the learning rate, and the sgm represents the sigmoid function given in Equation (14).

$$sgm(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

5. Online selection

For selecting the samples from streaming data, the Z-score method is used [45]. The instances with selection scores above a certain Z-score value are selected. The simple moving average (SMA) of the last k samples is used to determine the mean and variance [46]. The mean and variance are computed after every selection and used in Z-score calculation. The simple moving average calculation is given for the mean in Equation (15) and for the standard deviation in Equation (16). The Z-score value is given in Equation (17).

$$mean = \frac{1}{k} \sum_{i=1}^{k} score_i \tag{15}$$

$$\sigma = \frac{1}{k} \sum_{i=1}^{k} \sqrt{(score_i - mean)^2} \tag{16}$$

$$z = \frac{score - mean}{\sigma} \tag{17}$$

The selection approach does not use any threshold for selecting instances. Instead of a threshold, 10% of the 100 candidate samples with higher z-scores are selected. Through this approach, the divergence of z-score values across different methods is eliminated, the top-scored samples are selected, and the target vocabulary rate is preserved.

6. Evaluations

To evaluate the effects of active learning on Turkish word embeddings, we used semantic analogy tests of the Turkish word embedding evaluation dataset,[2] a supervised POS dataset [47], and sentiment analysis [48]. For each task, the words are selected from a large corpus. The large corpus contains 200 distinct sentences for each word in the training dataset of the target task. CBOW, SkipGram, and self-attention neural network models are used for word embedding extraction from the selected corpus [6,7]. Each active selection method is compared with a baseline, namely VocabSelect, where the sentences are selected randomly from the large corpus.

[2] Dataset: https://doi.org/10.17632/wc96srpx8m.1.

Table 1
Fixed parameters for active learning methods.

Parameter Name	Explanation	Value
Dictionary size	The number of maximum frequent words	80000
Stem length	Hopfield and Boltzmann networks	5
Hidden size	Hopfield, Boltzmann networks	500
Embedding size	Size of pre-trained embeddings	100
Subsequent embedding window	Window of the context	5
Committe size	Number of committees in language models	10
k	K-means cluster size	20
Window	Language models token window	20

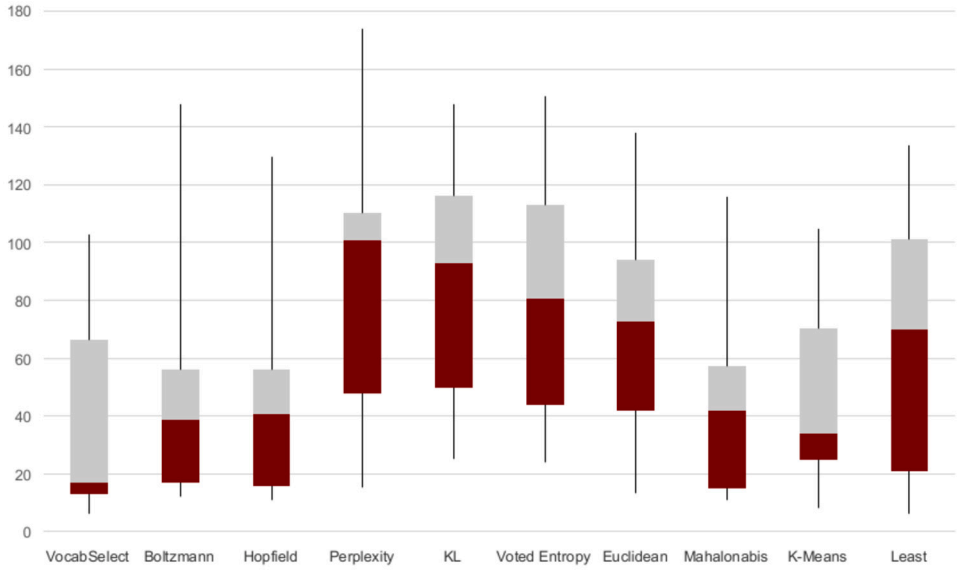

Fig. 4. Analogy results for all extraction methods and all sample sizes.

6.1. Experiments

The active learning methods are compared for varying numbers of sentence samples. Adapted methods are evaluated through the performance of the selected dataset in the tasks of analogy, POS tagging, and sentiment analysis. The selection methods include numerous parameters such as the number of committee members, number of cluster centers, window size, word embedding size, number of subsequent embeddings, dictionary size, stem length, and the number of hidden states. These selection criteria are fixed by several hand-tuned experiments on small samples of the dataset and are given in Table 1.

6.2. Prediction results

The evaluation results for the CBOW, SkipGram, and self-attention extraction models are provided in this section. Each model is used to evaluate the selection sizes of 1000, 5000, and 25000 samples. The scores for these tasks are given in the Tables of Appendix A. In the analogy task, cosine similarity is used to search the embedding of the analogy in the dictionary. If the target word is located in a dictionary, it is considered a true prediction; otherwise, it is considered a false prediction. The POS tagging and sentiment analysis are evaluated through a self-attention LSTM model with embedding layer fixed to the extracted embeddings from the selected sentences. The structure of the self-attention LSTM model is similar to the model presented in the study by Katrompas [7].

The findings presented in Appendix A and Fig. 4 indicate that the language models outperformed all other selection methods in terms of the number of true predictions. SkipGram outperforms CBOW and self-attention extraction models in terms of extraction model performance. The chosen sample size increases performance regardless of the extraction and selection process. This demonstrates that larger sample sizes are beneficial for analogy tasks.

In the evaluation, the F-Measure scores are reported for the test datasets. These scores are given in the tables of Appendix A. The method comparisons of the POS tagging task are given in Fig. 5. Based on the POS tagging scores, self-attention model performance is better than CBOW and SkipGram and the performance of the CBOW model is worse than skip-gram. There are cases where random selection performs better than other approaches. In Fig. 5, the results indicate that perplexity and KL scores relatively perform better than other selection methods. Compared to the results of the analogy task, the main performance improvement is observed

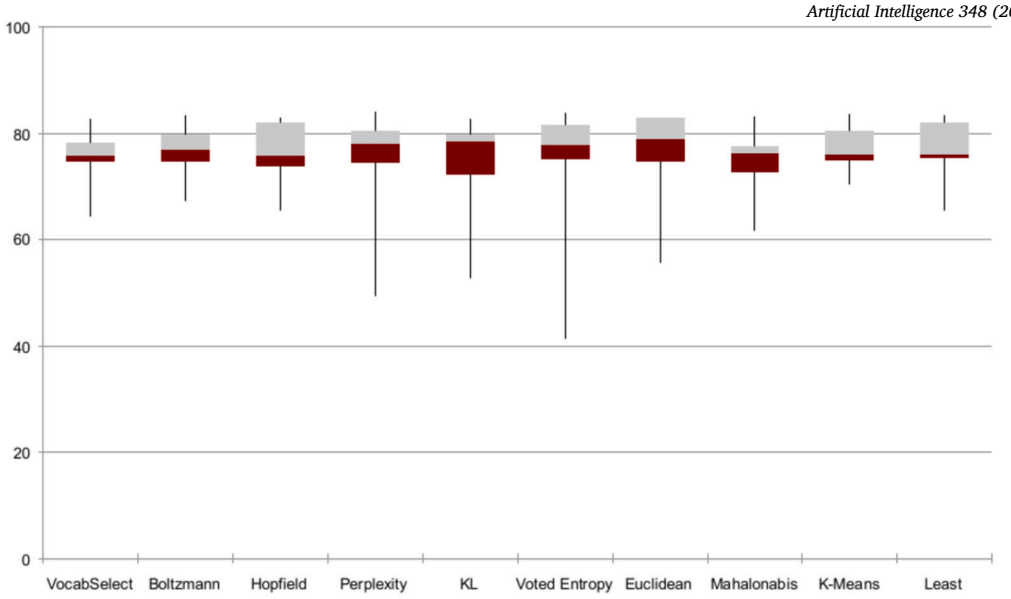

Fig. 5. POS tagging results for all extraction methods and all sample sizes.

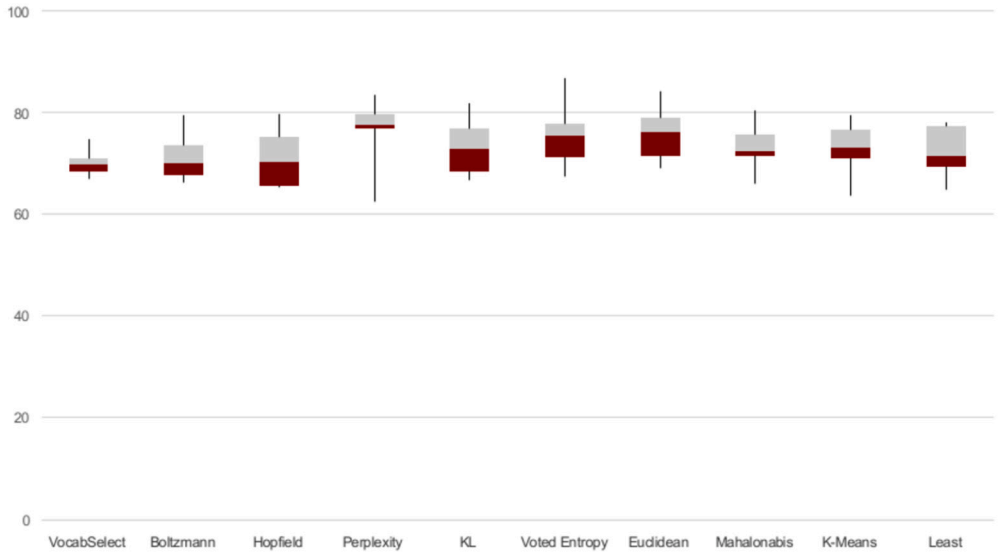

Fig. 6. Sentiment analysis results for all extraction methods and all sample sizes.

in Euclidean, and Least methods. These selection methods use global word embeddings to select the sentences and thus have more adaptation ability to the test dataset.

In Fig. 6, the sentiment analysis outcomes are presented. The results clearly show that perplexity- and Voted-Entropy–based selection methods achieve superior performance compared to other approaches. Notably, in Appendix A the SkipGram embeddings trained on Voted-Entropy–selected sentences exhibit approximately a 3% improvement over the baseline, confirming the advantage of uncertainty-driven criteria. Furthermore, the results highlight that even with a relatively small number of training sentences, substantial gains in performance can be obtained, underscoring the efficiency of informative sample selection for sentiment classification.

6.3. *Effects of sample size*

The selection sizes indicate that increasing the number of tokens with sentence length does not yield better prediction performance for almost all the compared methods. To test the impact of the density of the selected samples on performance, the scatter of the performances based on average token size is given in Figs. 7, 8, and 9.

A good performance comparison would answer whether increasing the number of samples improves performance. For all three tasks, the scatter plots show that the performance of each selection method generally decreases with increased average token size.

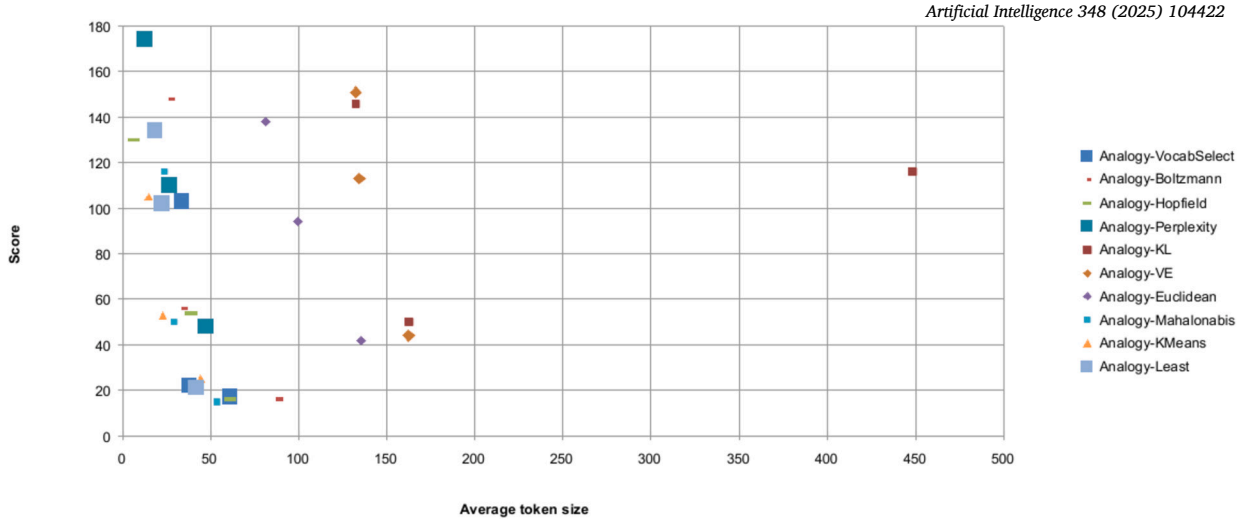

Fig. 7. Effects of average token size for the analogy task.

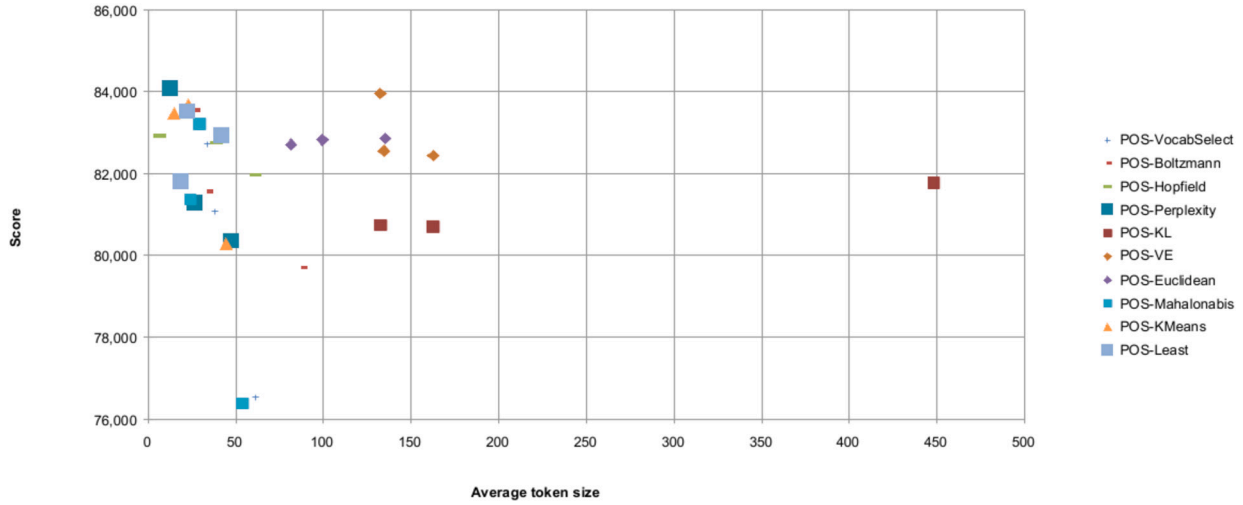

Fig. 8. Effects of average token size for the POS tagging task.

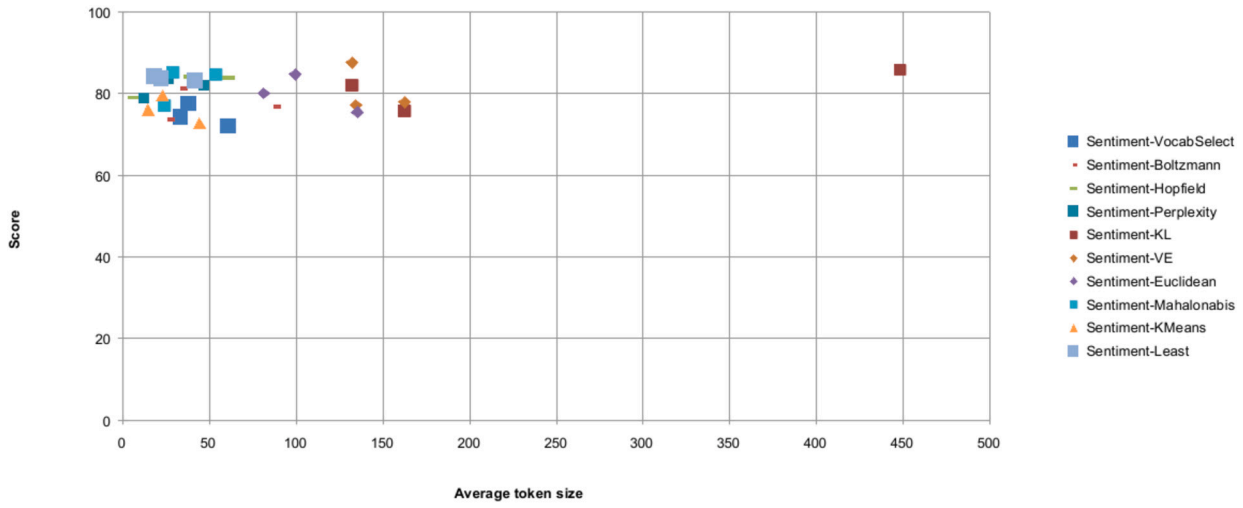

Fig. 9. Effects of average token size for the sentiment analysis task.

Table 2
Computational complexity of selection methods, where d denotes the dimensionality, n the number of tokens in a sentence, k the number of clusters in k-means, h the number of hidden states, and p the number of steps in contrastive divergence training.

Method	Selection Time T	Update Time	Notes
Perplexity	$O(n)$	$O(n)$	Token-pair lookups only.
Voted-Entropy	$O(Mn)$	$O(n)$	Sum entropies across M models.
KL-Divergence	$O(Mn)$	$O(n)$	Pointwise KL at each position.
Euclidean	$O(d)$	$O(d)$	With sparse BoW: proportional to nonzeros.
Mahalanobis	$O(d^2)$ (matrix–vector)	$O(d^2)$, inverse $O(d^3)$	Requires covariance inverse.
K-Means	$O(kd)$	$O(d)$	Distance per centroid.
Least Squares	$O(d)$	Refit: $O(Nd^2 + d^3)$	Dot product if trained offline.
Hopfield	$O(h^2)$	Expensive updates	Quadratic in states.
Boltzmann	$O(hd)$	$O(phd)$	Training dominates cost.

Table 3
Execution time in milliseconds for active learning methods applied to modeling and selecting a single sample, evaluated with datasets of 1000, 5000, and 25,000 sentences.

Method Name	Category	Efficiency		
		1000	5000	25000
K-Means	Clustering	55340.20	48957.32	48729.80
Least squares	Regression	17602.03	20283.90	62467.17
Voted Entropy	Language models	2783.98	3087.63	3750.69
KL Divergence	Language models	398.86	462.08	894.913
Perplexity	Language models	1169.94	1323.32	1890.89
Mahalonobis	Distance	37694.88	33611.28	36686.34
Euclidean	Distance	1268.98	2871.01	3132.38
Hopfield	Neural networks	463823.18	444684.95	441742.90
Boltzmann	Neural networks	429995.65	302210.11	441321.43

Table 4
Statistics of the sampled datasets for each method in selection of 25000 samples.

Method Name	Avg. Sentence Length	Distinct Tokens	Total Tokens
VocabSelect	194.153	26002.2	3348056.7
K-Means	183.667	26009.0	3167342.0
Least squares	133.557	26008.0	2297646.0
Voted Entropy	625.939	26008.0	2833711.0
KL Divergence	625.939	26009.0	2833715.0
Perplexity	126.387	26000.0	440176.0
Mahalonobis	190.652	26000.0	767382.0
Euclidean	404.929	26000.0	1781891.0
Hopfield	178.988	26000.0	2613820.0
Boltzmann	179.065	26005.0	3087146.0

The KL and Euclidean methods show a general improvement with the increased sample size in POS tagging and sentiment analysis. Mahalanobis distance method shows a linear improvement only in sentiment analysis. Increased average token size represents the data density per sample. It can be said that the average token size represents the density of the dataset. The results acknowledge that a larger vocabulary does not guarantee improvement in performance, but a dense vocabulary content yields generally better results.

In addition to empirical performance, a theoretical examination of computational efficiency highlights significant differences across methods. The computational complexity of each algorithm is given in Table 2 where n is the sentence length (tokens), d is the feature dimension, k is the number of clusters (K-Means), M is the number of committee LMs, p is the number of steps in contrastive learning and h is the number of hidden/state units. LM-based selectors reduce primarily to lookup and aggregation operations, scaling linearly with sentence length and, in the case of ensembles, with the number of models. By contrast, geometric approaches such as Mahalanobis distance require matrix operations whose cost grows quadratically or cubically with feature dimensionality. Clustering methods such as K-Means scale with the number of centroids, while regression-based methods incur low inference cost but high retraining complexity. Finally, energy-based models demand quadratic or bilinear computations, rendering them computationally expensive for large-scale streaming tasks. These asymptotic properties explain the observed wall-clock trends and underscore the practicality of LM-based selectors for both accuracy and throughput in large corpora. (See Tables 3 and 4.)

7. Discussion

Methods grounded in language model uncertainty – Perplexity, Voted-Entropy, and KL-Divergence – consistently outperformed clustering-, distance-, regression-, and energy-based alternatives. Their strength lies in capturing order-sensitive information: they measure how much a sentence "surprises" a model, thus quantifying its informativeness. Perplexity identifies sentences the baseline LM finds atypical, Voted-Entropy aggregates disagreement across models, and KL-Divergence measures distributional shifts relative to a committee average. By focusing on sentences that maximize predictive uncertainty, these methods align closely with the objective of reducing model error. Empirically, this translated into superior scores across analogy and POS tagging tasks: for example, perplexity- and KL-based selection yielded significantly higher true predictions and F1-scores than Euclidean or regression-based selection (Appendix Tables A.5–A.7). Their consistent gains highlight that uncertainty-driven criteria directly enhance representation learning by forcing models to internalize complex syntactic and semantic patterns.

Distance metrics, clustering, regression, and energy-based models underperformed because they neglect sequential dependencies. Euclidean and Mahalanobis distance treat sentences as unordered vectors, promoting lexical diversity but often selecting semantically redundant material. Only at larger sample sizes did Euclidean become competitive, when broad lexical spread began to overlap with informational diversity. Mahalanobis narrowed the gap in sentiment analysis, but only when covariance aligned with task features (e.g., polarity axes). K-Means clustering ensured space coverage but overlooked linguistically salient sentences outside large clusters. Regression residuals flagged feature outliers without guaranteeing syntactic or semantic novelty, explaining their weaker analogy results. Finally, Hopfield and RBM models minimized energy rather than predictive uncertainty, producing selections poorly aligned with next-token prediction while imposing extreme computational costs. These trends confirm that approaches not tied to sequence modeling objectives contribute less to embedding quality.

The value of each method depended on dataset demands. Tasks requiring fine-grained local dependencies – analogy and POS tagging – benefited most from LM-based selectors, which identified sentences rich in contextual cues. For instance, self-attention POS tagging achieved F1 scores above 83 with KL- and entropy-based selection, clearly outperforming random or regression baselines. In contrast, sentiment analysis, which relies more on global lexical coverage, narrowed the performance gap: Euclidean and regression methods improved substantially here, consistent with their broader coverage bias. Mahalanobis also showed linear improvement with larger sample sizes in sentiment classification, reflecting its ability to capture global covariance when aligned with task-level distinctions. These findings demonstrate that while LM selectors are most robust, other methods can occasionally contribute under task conditions favoring lexical breadth over contextual depth.

Corpus density and domain alignment shaped outcomes as much as method choice. Larger corpora generally improved results, but gains were uneven: analogy tasks benefited from expanded vocabulary coverage, while POS and sentiment tasks gained from denser sampling of common structures. Perplexity, which often picked shorter, information-dense sentences, excelled in POS tagging but sometimes underperformed in sentiment tasks if sentiment-bearing vocabulary was missed. Conversely, KL and Voted-Entropy frequently selected long, complex sentences, which maximized model disagreement but risked domain mismatch when test sets contained shorter, simpler forms. These observations highlight that effectiveness depends not only on method but also on how well the distribution of selected data matches downstream requirements.

Together, these results reinforce the principle that "quality over quantity" governs effective unsupervised selection. Perplexity-based selection achieved high downstream accuracy with fewer total tokens, confirming that information density is more valuable than corpus size. KL and entropy-based methods balanced informativeness with diversity, though their bias toward longer sentences must be managed. Ultimately, contextual uncertainty, not raw size, proved decisive for embedding improvement. For practitioners, this suggests aligning method choice with task needs: entropy-driven selectors for syntactically sensitive tasks, diversity-ensuring methods when broad lexical coverage matters. Future work could explore hybrid approaches that first guarantee coverage then refine selection by uncertainty. The evidence confirms that active learning methods are effective for principled reasons, and their interaction with dataset properties provides both theoretical grounding and practical guidance for scalable corpus construction.

8. Conclusion

This study systematically evaluated active learning strategies for unsupervised sentence selection in Turkish—spanning clustering, regression, distance-based metrics, neural energy models, and language model–driven criteria. Across analogy, POS tagging, and sentiment analysis, uncertainty-based methods (perplexity, KL divergence, voted-entropy) consistently delivered the strongest gains at comparatively low computational cost. We observed task-specific dynamics: analogy benefited from broader vocabulary coverage, whereas POS and sentiment improved most with denser, context-rich samples—indicating that data density is more consequential than raw corpus size. Neural energy models, despite their representational appeal, yielded limited returns due to high computational overhead and weak alignment with language modeling objectives.

Building on these findings, we advocate a practical recipe for corpus construction in morphologically rich settings: prioritize uncertainty-based selection; favor dense, contextually rich sentences over indiscriminate scaling; and introduce lightweight pre-filtering to reduce downstream compute. In particular, compact scoring modules that estimate sentence utility—grounded in perplexity, entropy, or divergence—can serve as front-end filters to screen candidates before expensive model inference.

Looking forward, integrating these scoring functions as adaptive filtering layers within large models is a promising direction. Such layers can operate in tandem with prediction, evolving as the model learns to dynamically reweight what counts as "informative," thereby preserving predictive performance while systematically curbing training cost. Collectively, these recommendations chart a scalable, cost-effective path for high-quality corpus construction in morphologically rich—and often low-resource—languages.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Results

Table A.5
True word counts for analogy task.

Extraction	VocabSelect			Boltzmann			Hopfield			Perplexity			KL		
	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K
CBOW	13	11	66	17	30	89	16	41	76	15	101	167	25	93	148
SkipGram	6	13	103	16	56	148	11	54	130	**48**	110	**174**	50	**116**	146
Self-attention	17	22	96	12	39	47	11	37	56	38	64	103	33	72	99
	Voted Entropy			Euclidean			Mahalonabis			K-Means			Least		
	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K
CBOW	24	92	151	13	88	98	11	33	110	16	34	79	21	62	101
SkipGram	44	113	151	42	94	138	15	50	116	8	53	105	6	102	134
Self-attention	31	79	81	38	66	73	11	42	57	25	30	70	18	70	78

Table A.6
The f-measure scores for POS tagging task.

Extraction	VocabSelect			Boltzmann			Hopfield			Perplexity			K-Means		
	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K
CBOW	72.190	73.692	64.312	73.110	74.662	67.202	73.272	73.869	65.339	74.539	63.379	70.024	75.048	74.400	70.236
SkipGram	74.675	74.883	68.158	75.852	76.992	78.334	75.508	76.188	75.788	75.246	78.112	79.786	75.438	76.199	77.210
Self-attention	76.534	81.069	81.720	79.714	81.562	83.552	81.966	82.752	82.923	80.345	81.275	**84.068**	80.281	**83.690**	83.482
	KL			Voted Entropy			Euclidean			Mahalonabis			Least		
	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K
CBOW	62.397	72.684	73.501	75.151	63.264	61.284	74.736	75.461	61.982	72.696	70.441	61.620	75.514	73.646	65.287
SkipGram	75.570	78.471	79.680	76.255	77.992	79.424	75.841	78.371	80.661	72.791	76.6780	77.473	75.502	76.083	77.324
Self-attention	80.700	82.762	79.737	82.635	81.550	83.962	**82.857**	82.821	82.710	76.385	83.217	81.365	82.953	83.515	81.800

Table A.7
The f-measure scores for sentiment analysis task.

Extraction	VocabSelect			Boltzmann			Hopfield			Perplexity			K-Means		
	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K
CBOW	71.398	73.361	70.620	73.168	81.262	73.754	72.248	78.444	65.162	81.904	62.335	76.735	72.758	76.189	63.677
SkipGram	61.997	73.039	74.293	76.746	72.026	85.340	83.975	84.828	70.709	77.564	83.582	68.791	63.599	69.556	66.950
Self-attention	69.382	77.452	74.259	75.038	78.555	72.048	70.038	78.537	79.029	77.401	77.933	77.903	71.309	77.903	76.127
	KL			Voted Entropy			Euclidean			Mahalonabis			Least		
	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K	1K	5K	25K
CBOW	72.420	80.218	82.114	72.805	75.639	80.458	72.422	79.035	80.125	77.557	75.251	76.525	74.549	78.535	64.891
SkipGram	71.930	**85.835**	69.428	77.907	68.087	**87.617**	69.567	84.734	70.965	84.763	85.208	66.697	83.167	83.620	84.142
Self-attention	75.728	78.019	80.434	69.630	77.243	81.254	75.450	78.227	78.276	76.660	78.667	77.069	76.747	75.842	73.620

Appendix B. Neural network model

In the evaluation of word embeddings, we employed two downstream tasks—part-of-speech (POS) tagging and sentiment analysis—each modeled with neural architectures tailored to their specific requirements (see Figs. B.10 and B.11). Both tasks rely on pre-trained word embeddings as their lexical foundation. Input tokens are mapped to fixed representations constructed from averaged n-gram embeddings, ensuring that subword information is incorporated into the vector space. These embeddings remain static during supervised training to allow a fair comparison across different embedding methods, including Skip-Gram, CBOW, and self-attention LSTM embeddings.

For sentiment analysis and for extracting word embeddings, the architecture combines recurrent and attention mechanisms. Instead of a bidirectional LSTM, a single LSTM layer is employed to encode the input sequence. To address the limitations of plain LSTMs, particularly their reduced sensitivity to earlier tokens in long sequences, we integrate a self-attention layer over the LSTM outputs. This mechanism computes a weighted sum of all hidden states, selectively emphasizing tokens that are most informative for sentiment classification (e.g., polarity shifters such as negations). The resulting context vector is passed to a dense output layer with a softmax activation, producing sentence-level polarity predictions.

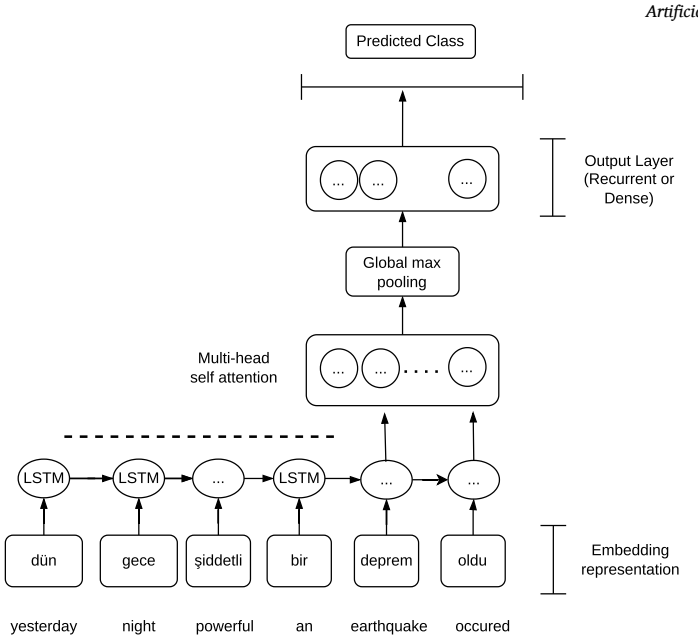

Fig. B.10. A LSTM with multi-head attention layer for sentiment analysis.

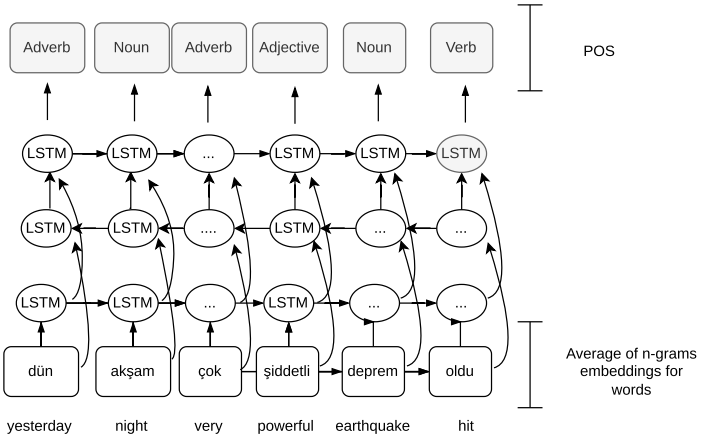

Fig. B.11. A Bi-LSTM for POS tagging.

For POS tagging, the model architecture is centered on a two-layer bidirectional Long Short-Term Memory (BiLSTM) network. This design captures sequential dependencies from both left and right contexts, thereby improving the modeling of morphosyntactic patterns across a sentence. Each token's embedding is transformed by the BiLSTM into contextually enriched hidden states, which are directly used for token-level classification. The output layer produces a categorical distribution over POS tags for each token, making the architecture well-suited for sequence labeling tasks.

Data availability

I have shared the link to my data.

References

[1] Y. Fu, X. Zhu, B. Li, A survey on instance selection for active learning, Knowl. Inf. Syst. 35 (2013) 249–283.
[2] B. Miller, F. Linder, W.R. Mebane, Active learning approaches for labeling text: review and assessment of the performance of active learning approaches, Polit. Anal. 28 (4) (2020) 532–551.
[3] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[4] F. Olsson, A literature survey of active machine learning in the context of natural language processing, 2009.
[5] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[6] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inf. Process. Syst. 26 (2013).
[7] A. Katrompas, V. Metsis, Enhancing lstm models with self-attention and stateful training, in: Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys), vol. 1, Springer, 2022, pp. 217–235.
[8] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, 2016.
[9] Z. Bodó, Z. Minier, L. Csató, Active learning with clustering, in: Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010, JMLR Workshop and Conference Proceedings, 2011, pp. 127–139.
[10] D.A. Cohn, Z. Ghahramani, M.I. Jordan, Active learning with statistical models, J. Artif. Intell. Res. 4 (1996) 129–145.
[11] S. Kee, E. Del Castillo, G. Runger, Query-by-committee improvement with diversity and density in batch active learning, Inf. Sci. 454 (2018) 401–418.
[12] S. Xiong, J. Azimi, X.Z. Fern, Active learning of constraints for semi-supervised clustering, IEEE Trans. Knowl. Data Eng. 26 (1) (2013) 43–54.
[13] Y. Leng, X. Xu, G. Qi, Combining active learning and semi-supervised learning to construct svm classifier, Knowl.-Based Syst. 44 (2013) 121–131.
[14] B. Xie, L. Yuan, S. Li, C.H. Liu, X. Cheng, G. Wang, Active learning for domain adaptation: an energy-based approach, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 8708–8716.
[15] A. Sethy, P.G. Georgiou, S. Narayanan, Selecting relevant text subsets from web-data for building topic specific language models, 2006, pp. 145–148.
[16] W. Hu, W. Hu, N. Xie, S. Maybank, Unsupervised active learning based on hierarchical graph-theoretic clustering, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 39 (5) (2009) 1147–1161.
[17] C. Li, K. Mao, L. Liang, D. Ren, W. Zhang, Y. Yuan, G. Wang, Unsupervised active learning via subspace learning, 2021, pp. 8332–8339.
[18] D. Hakkani-Tur, G. Tur, M. Rahim, G. Riccardi, Unsupervised and active learning in automatic speech recognition for call classification, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, IEEE, 2004, I–429.
[19] B. Antoine, S. Ertekin, J. Weston, L. Bottou, Fast kernel classifiers with online and active learning, J. Mach. Learn. Res. 6 (2005) 1579–1619.
[20] C. Li, H. Ma, Y. Yuan, G. Wang, D. Xu, Structure guided deep neural network for unsupervised active learning, IEEE Trans. Image Process. 31 (2022) 2767–2781.
[21] H. Ma, C. Li, X. Shi, Y. Yuan, G. Wang, Deep unsupervised active learning on learnable graphs, IEEE Trans. Neural Netw. Learn. Syst. (2022), https://doi.org/10.1109/TNNLS.2022.3190420.
[22] C. Wan, F. Jin, Z. Qiao, W. Zhang, Y. Yuan, Unsupervised active learning with loss prediction, Neural Comput. Appl. (2021) 1–9.
[23] Y. Yang, Z. Ma, F. Nie, X. Chang, A.G. Hauptmann, Multi-class active learning by uncertainty sampling with diversity maximization, Int. J. Comput. Vis. 113 (2015) 113–127.
[24] H. Trittenbach, A. Englhardt, K. Böhm, An overview and a benchmark of active learning for outlier detection with one-class classifiers, Expert Syst. Appl. 168 (2021) 114372.
[25] B. Settles, Active learning literature survey, 2009.
[26] D. Yoo, I.S. Kweon, Learning loss for active learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 93–102.
[27] J. Kremer, K.S. Pedersen, C. Igel, Active learning with support vector machines, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 4 (2018) 313–326, uncertainty sampling by Lewis and Gale.
[28] T. Scheffer, C. Decomain, S. Wrobel, Active hidden Markov models for information extraction, in: Advances in Intelligent Data Analysis: 4th International Conference, Proceedings, IDA 2001 Cascais, Portugal, September 13–15, 2001, vol. 4, Springer, 2001, pp. 309–318.
[29] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 2008, pp. 1070–1079.
[30] J. Han, S. Kang, Active learning with missing values considering imputation uncertainty, Knowl.-Based Syst. 224 (2021), https://doi.org/10.1016/j.knosys.2021.107079.
[31] H. Anahideh, A. Asudeh, S. Thirumuruganathan, Fair active learning, Expert Syst. Appl. 199 (2022) 116981.
[32] R. Caramalau, B. Bhattarai, T.-K. Kim, Sequential graph convolutional network for active learning, 2021.
[33] M. Wang, F. Min, Z.-H. Zhang, Y.-X. Wu, Active learning through density clustering, Expert Syst. Appl. 85 (2017) 305–317.
[34] P. Liu, L. Wang, R. Ranjan, G. He, L. Zhao, A survey on active deep learning: from model driven to data driven, ACM Comput. Surv. 54 (10s) (2022) 1–34.
[35] J. Lu, P. Zhao, S.C. Hoi, Online passive-aggressive active learning, Mach. Learn. 103 (2016) 141–183, https://doi.org/10.1007/s10994-016-5555-y.
[36] X. Zhan, H. Liu, Q. Li, A.B. Chan, A comparative survey: benchmarking for pool-based active learning, in: IJCAI, 2021, pp. 4679–4686.
[37] X. Zhu, P. Zhang, X. Lin, Y. Shi, Active learning from data streams, in: Seventh IEEE International Conference on Data Mining (ICDM 2007), IEEE, 2007, pp. 757–762.
[38] M.R. Bouguelia, Y. Belaïd, A. Belaïd, An adaptive streaming active learning strategy based on instance weighting, Pattern Recognit. Lett. 70 (2016) 38–44, https://doi.org/10.1016/j.patrec.2015.11.010.
[39] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognit. 36 (2) (2003) 451–461.
[40] D. Wu, C.-T. Lin, J. Huang, Active learning for regression using greedy sampling, Inf. Sci. 474 (2019) 90–105.
[41] D. Cacciarelli, M. Kulahci, J.S. Tyssedal, Stream-based active learning with linear models, Knowl.-Based Syst. 254 (2022), https://doi.org/10.1016/j.knosys.2022.109664.
[42] C. Marullo, E. Agliari, Boltzmann machines as generalized Hopfield networks: a review of recent results and outlooks, Entropy 23 (1) (2020) 34.
[43] A. Fischer, C. Igel, Empirical analysis of the divergence of Gibbs sampling based learning algorithms for restricted Boltzmann machines, in: ICANN, vol. 3, vol. 6354, 2010, pp. 208–217.
[44] G.E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 599–619.
[45] I. Markov, A. Arampatzis, F. Crestani, Unsupervised linear score normalization revisited, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 1161–1162.
[46] I. Svetunkov, F. Petropoulos, Old dog, new tricks: a modelling view of simple moving averages, Int. J. Prod. Res. 56 (2018) 6034–6047, https://doi.org/10.1080/00207543.2017.1380326.
[47] B. Say, D. Zeyrek, K. Oflazer, U. Özge, Development of a corpus and a treebank for present-day written Turkish, in: Proceedings of the Eleventh International Conference of Turkish Linguistics, Eastern Mediterranean University, 2002, pp. 183–192.
[48] A. Ucan, B. Naderalvojoud, E.A. Sezer, H. Sever, Sentiwordnet for new language: automatic translation approach, in: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2016, pp. 308–315.