

Intermediate Steps REPORT

Formal Semantics, WS 17/18, WSI Project
Chernenko Tatjana, Toyota Utaemon

Milestone	Step	Description	Comments	Person
Preparation	Read papers	Read the description paper (http://aclweb.org/anthology/S/S13/S13-2035.pdf); sense2vec paper (https://arxiv.org/abs/1511.06388); sent2vec paper (https://github.com/epfml/sent2vec); evaluation paper (http://www.aclweb.org/anthology/J13-3008)		Tatjana, Utaemon
	Discuss the idea	Discuss WSD and WSI tasks; look for available Datasets and Sense Repositories, tools for automatic labeling, etc.		Tatjana, Utaemon
	Find resources	Find a Training Corpus	Wikipedia 2017	Tatjana, Utaemon
	Install dependencies	Install fasttext, gensim, scikit-learn, sense2vec, sent2vec, word2vec; download trial data and evaluator script, get acquainted with the data		Tatjana, Utaemon
	Organisation	Create a gitlab repository; create a TO-DO list for the project; intermediate steps report	https://gitlab.cl.uni-heidelberg.de/semantik_project/wsi_chernenko_toyota	Tatjana, Utaemon
Implementation	Preprocess Wikipedia Dataset	Download english Wikipedia Dump from 20th August 2017	enwiki-20170820-pages-articles-multistream.xml.bz2	Utaemon
		Install wikiExtractor and extract the Wikipedia text from XML		
		Preprocess text to remove unnecessary quotations ('. '), brackets and comments to receive plain text		
		Create a file with whole Wikipedia text which contains one sentence per line		
	Train sent2vec unigramms and bigramms models	Creating new uni-gram and bi-gram models over the preprocessed Wikipedia text		Utaemon
		Models include a total of 321 million words and 4518148 number of words		
		Size of Unigram model: 25.4GB; Bi-gram model: 36.6GB		
	Implement the baseline	Implement the baseline for the WSI task (read input files; preprocess snippets; devide snippets into topics; create a vector representation of snippets; create a data structure for saving snippets, IDs, topic names, etc.; use vector mixture model for compositional representation of the snippets; apply a clustering algorithm; write the clustering results to output files, using the right format)		Tatjana
	40 experiments	Try to improve the baseline and find the best model using word embeddings: run the experiments with different combinations of:		Tatjana

Implementation	40 experiments	- preprocessing steps (tokenization, punctuation removal, capitalization removal, POS-tagging, stop-words removal)		Tatjana
		- language models (sense2vec; sent2vec with 3 pre-trained models, sent2vec with 2 self-trained models; word2vec)		
		- compositional semantics (BOW summarization, tuned BOW summarization with weighted vectors)		
		- default and given number of clusters		
		- clustering algorithms (KMeans, MeanShift, Affinity Propagation, Spectral Clustering, Agglomerative Clustering, cosine similarity, cosine similarity with min factor)		
	Evaluation	Evaluate 40 outputs of the experiments and baseline, compare the influence of different steps and features on the performance		Tatjana
	Discussion	Discuss the results, choose the best performing model		Tatjana, Utaemon
Post-processing	Implement the system	Work on the model, improve the code, add command-line arguments, etc.		Tatjana
	Create a performance table	Create a big table with evaluation results of the baseline and experiments		Tatjana, Utaemon
	Use test data	Apply the system on the test data, save output results		Tatjana, Utaemon
	Documentation	Create README.md files and running instructions		Tatjana, Utaemon
	Intermediate steps report	Describe all the processing steps		Tatjana, Utaemon
	Report	Write a project report		Tatjana, Utaemon