Papers	Feature Representation	Algorithm	Pre-processing	Post-processing	Dataset	Evaluation Metrics	Results
Effects of Pre- and Post-Processing on type-based Embeddings in Lexical Semantic Change Detection	Using SGNS to create type-based word representations in combination with different alignment methods Corpora: English and German datasets provided by SemEval-2020 Task 1 Subtask 2	Create semantic word representation Align across corpora Measuring differences between aligned representations	German - Vector Initialization on one corpus then using the learned word and context vectors to initialize the model for training on the other corpus English - Orthogonal Procrustes Removing stopwords Stemming words	Similarity Over Time plus Stacking for both languages overall Mean Centering plus Principal Component Removal for even better performance for English	English and German datasets provided by SemEval-2020 Task 1 Subtask 2	Spearman's rank-order correlation coefficient. Rank the list of target words according to their sense divergence from 0 to 1 (no change to complete change) Compared to gold standard data based on human judgement	SOT + STA => moderate performance increase for both german and english MC+PCR => larger performance increase for english, not much for german Neither were able to find a reliable parameter that performed well across the board Well-performing parameter values are dependent on the underlying matrix
A Framework for Analyzing Semantic Change of Words across Time	5-grams of each word with weight value of total number of co-occurring words 5-grams of each word with positional relevance LSA (Latent Semantic Analysis), used to represent underrepresented words more and overrepresented words less. Corpora: Google Books 5-gram dataset COHA	Construct word vector representation for each context in each decade Compare the context in the last decade with the ones in the previous decades using cosine similarity	Remove all context words less than 1% frequency of most frequent word Remove digits and non-words Lowercase all words Remove stopwords using NLTK list	None	Google Books 5-gram dataset COHA	Cosine similarity per decade graphical representations	Provided a view of the change of words and word pairs semantics over time for easy comparisons and contrasts. Allows for easier access to when words came into and out of meaning.
Deep Convolution Neural Networks for Twitter Sentiment Analysis	GloVe model (Global Vectors for word representation) 200-dimensional vector Corpora: 20 Billion tweets	n x k representation of tweet into first layer of dcnn Max pooling and map vector to fixed lengths Final layer is a dropout with softmax output	Remove all non-ASCII and non-English characters Remove all URL links Remove all numbers Replace negative references Expand acronyms and slang Remove all stopwords Replace emoticons and emojis with text Tokenization using Tweet-NLP	None	STSTd SE2014 STSGd SSTd SED	10-fold validation for each dataset Replicate cross validation experiments 100 times for each dataset. Finally the average of accuracy, F1-Measure, precision, and recall each dataset is used	Faster than any method before it and does not rely on human input (labelling). Accuracy of 87.62% Maximum improvement of 19.14% Minimum improvement of 3.68% over baseline Highest precision, recall, and F1 score over all other methods
A Bayesian Model of Diachronic Meaning Change	Ngram of target words with window of +-5 words annotated with year of origin Corpora: DATE Corpus	Uses SCAN (Captures how a word's senses evolve over time) Conflate all documents from the same time period and infer a temporal representation of the target word per time period. Each representation is a multinomial distribution over the word senses. Each model infers a precision parameter, which controls the extent to which each sense prevails for any word over time. Predict time period by using distribution and update model.	Remove stopwords and function words	None	Google Books (bigram) for 1960s and 1990s, with annotations using a 4-point ordinal scale	Spearman's rank-order correlation coefficient.	Performed on par with the best published results without any extensive feature engineering or task specific tuning
Lexical Semantic Relatedness for Twitter Analytics	Create a twitter word dependency graph using the conditional dependency between tweet tokens (userID, tweetID, tweet, tokenID, token, isHashtag) and the co-occurence count between tweets. Corpora: Cheng. Z (2010), 8.7 Million tweets	Create tweet tokens Calculate co-occurrences Find conditional dependencies Create dependency graph Calculate semantic similarity between TSSR using cosine similarity Evaluate using gold standard-base evaluation, compare against best performing semantic relatedness technique (ESA) on application specific problems, and use human subjects to determine to evaluate non-dictionary words	Removing stop words and stemming words	None	Golden Standard: WordSimilarity, RG-65, and MC-30	Golden Standard: Spearman's Rank-Order Correlation Coefficient and Mean Absolute Error ESA: Compared TSSR to ESA, similarity of a tweet to a query is calculated as the sum of semantic relatedness between query terms and tweet body Human Subjects: Five level Likert-type scale for 25 words for a given hashtag, asked to rate the relatedness of words to hashtags	Achieved higher scores among human subjects, ICC score, and a tweet search, which is a more realistic approach and is also important in areas of research involving searching techniques. Did not do as well in gold standard-based evaluation because twitter is not a place of proper grammar and spelling.
Modelling the Semantic Change Dynamics using Diachronic Word Embeddings	Diachronic Word Embeddings - word embeddings for each time-period then aligning all of them temporally. Each word is a set of 20 continuous 300th dimensional vectors, 1 vector for each decade Corpora: Google Books N-gram datasets	Recurrent Neural Network - Regression problem Used LSTM (Long short-term memory) architecture, designed to effectively deal with vanishing gradient that RNNs suffer from. Trained from diachronic word embeddings, predicts word embeddings for next time step. Uses cosine similarity to base closeness of meanings between words to determine word embedding.	Used word2vec algorithm w/ Skip-Gram w/ negative sampling variant. Used to create the sense vector for each word.	None	None	Tested: Top 1000, 5000, 10000, and 20000 words. Used a 10-fold cross validation method. A correct prediction was calculated by measuring the cosine similarity of the predicted vector and a word from the vocabulary. If the word is the nearest semantic neighbor then the prediction is correct else it is false.	Vocabulary Size : Accuracy: 1000 : 91.7% 5000 : 86.1% 10000 : 71.4% 20000 : 52.2% Better performance for smaller vocabulary sizes

EmbLexChange at SemEval-2020 Task 1: Unsupervised Embedding-based Detection of Lexical Semantic Change	Word embeddings, separate for each time period, creates continuous vector representations for each time period Create word profiles for fixed points that do not exhibit significant semantic change Corpora: CCOHA DTA, BZ, ND LatinISE KubHist	Train embeddings using the Skip-Gram architecture for each language and time periods. Set window size = 7 and embedding size = 100. Prepare a set of frequent words that exists in both time periods, then filter the set by their ratio of normalized frequency. Apply accuracy metric for binary detection and for ranking order metric use the Spearman rank-order correlation coefficient and Kendell-τ.	Train embeddings using the Skip-Gram architecture for each language and time periods. Set window size = 7 and embedding size = 100. Prepare a set of frequent words that exists in both time periods, then filter the set by their ratio of normalized frequency.	None	CCOHA (English) DTA, BZ, ND (German) LatinISE (Latin) KubHist (Swedish)	Apply accuracy metric for binary detection and for ranking order metric use the Spearman rank-order correlation coefficient and Kendell-τ.	Achieved second place at the SemEval-2020 Task 1 with an average accuracy 0.686 compared to the first place accuracy of 0.687.
Diachronic word embeddings and semantic shifts: a survey [The paper focuses on the evolution and current state of lexical semantic change, not proposing any methods itself]	Time tensor with random indexing, word frequency distributions, distributional word representations, word embeddings Corpora: Google Ngrams Corpus NYT Corpus COHA Corpus Gigaword Corpus	Mean Shift Model, detections of bursts of words Word2Vec w/ dynamic Skip-Gram, possible to learn word embeddings across time jointly.	None	None	None	None	Provided a wide view of the field of diachronic lexical semantic change and the older methods to the newer methods with different types of word representations, corporas, and methods of detecting semantic change.
A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains	Bag-of-Words based: L/P and Lall (explained in pre-processing), with window sizes of 2, 5, and 10. Lall is used for DURel dataset L/P is used for SURel dataset Corpora: DTA (Deutsches Textarchiv) Cook (Recipes) SDEWAC (General Language)	Apply pre-processing SGNS (Skip-Gram w/ Negative Sampling) with Orthogonal Procrustes Measured by Cosine Similarity Ranked based on Spearman's rank-order correlation coefficient	Removed words below a frequency threshold t. COOK, t=2 DTA 18, t=25 DTA 19, t=37 SDEWAC, t=97 Punctuation removed and lemmatized, Lall Only context words, punctuation removed, lemmatized, and POS-tagging, L/P	None	DURel (Diachronic Usage Relatedness) SURel (Synchronic Usage Relatedness)	Spearman's rank-order correlation coefficient.	Best combination was Skip-Gram w/ Negative Sampling, Orthogonal Procrustes, and Cosine similarity, with window sizes 10 for Lall and 2 for L/P, and parameters k=1, and t=0/0.001 respectively.
Time-Out: Temporal Referencing for Robust Modeling of Lexical Semantic Change	Low-dimensional embeddings for SGNS High-dimensional sparse vectors for PPMI Corpora: COHA COCA	Apply pre-processing Tested Skip-Gram with Negative Sampling and Positive Pointwise Mutual Information models with normal alignment, creating word context pairs, and Temporal Referencing, avoids alignment, creates time pairs for words and creates one vector space representing all words. Measure cosine similarity between representations for lexical semantic change Calculate the difference between the average cosine distance between the genuine and shuffled conditions	Lower-cased all tokens, minimum frequency 100	None	WSC TestSet	Calculate the difference between the average cosine distance between the genuine and shuffled conditions The higher the value, the less noisy and better the model.	SGNS model trained with temporal referencing has significantly less noise than standard alignment. Alignment: 0.033, TR: 0.059 Difference: 0.026 The PPMI model where no alignment is necessary, temporal referencing also reduces the noise level, not as much as for the SGNS model. Alignment: 0.028, TR: 0.033 Difference: 0.005
Diachronic Usage Relatedness (DURel): A Framework for the Annotation of Lexical Semantic Change	Synchronic polysemy annotation to diachronic changes annotations	Calculate relatedness between words at certain time periods, then determine semantic proximity and thus determines polysemy.	None	Five native speakers of German rated 1,320 pairs of words on the 4-point scale to determine relatedness	Creates a framework for use in diachronic lexical semantic change with polysemy annotations. 60 use pairs per word and 1,320 use pairs with 22 target words	Spearman's rank-order correlation coefficient.	Average correlation score, between 0.55 and 0.62
A State-of-the-Art of Semantic Change Computation	Word vectors, word embeddings SVD, SGNS, PPMI, SCAN Corpora: Google Books Ngram Corpus COHA Google Books Syntactic Ngram Corpus Helsinki Corpus of English Texts etc.	Summarizes current approaches to the field of diachronic semantic change	Pre-process words by removing punctuations, unnecessary symbols, and lowercasing all tokens	None	SiBol/Port Dataset BNC–ukWaC Dataset	Pairwise Pearson correlations Dictionary comparison w/ Cosine similarity Sense mapping	SGNS and SVD perform differently on different corpus and PPMI performs worse than either of those two. Adding different characterizations to the word senses leads to different results, SCAN method, therefore there is no guarantee that SGNS or SVD are the better methods until more testing is done on the larger evaluations.