Recognizing Textual Entailment (RTE)

Master's Thesis Colloquium Talk

Madhumita
Supervised by:
Prof. Dr. Günter Neumann
Prof. Dr. Dietrich Klakow

Department of Computational Linguistics
Saarland University

Outline

Outline

What is Text Entailment?

- "T" entails "H", if a human reading them can infer H from T.
- Example:
  - Text(**T**): Google files for its long awaited IPO.
  - Hypothesis(**H**): Google goes public.

Outline

Why is it important?

- Provides semantic relation between pairs of text.

Why is it important?

- Provides semantic relation between pairs of text.
- Some use cases:
  - Information Retrieval: Query entailed in retrieved documents.
  - Question Answering: Reformulated search query entailed in answer.
  - Answer scoring: Student answer entailed in expected answer.
  - Machine Translation: Bidirectional, cross-lingual entailment between source and target.

Outline

Task complexity

- Highly complex due to semantic nature.

Task complexity

- Highly complex due to semantic nature.
- Low surface level match, but entailment.
  - Text(**T**): This incident has left me speechless. I can't be any happier at the turn of events.
  - Hypothesis(**H**): I'm elated due to the latest development.

Task complexity

- Highly complex due to semantic nature.
- Low surface level match, but entailment.
  - Text(**T**): This incident has left me speechless. I can't be any happier at the turn of events.
  - Hypothesis(**H**): I'm elated due to the latest development.
- High surface level match, but non-entailment.
  - Text(**T**): I am anything but a coffee addict.
  - Hypothesis(**H**): I am a coffee addict.

Task complexity

- Highly complex due to semantic nature.
- Low surface level match, but entailment.
  - Text(**T**): This incident has left me speechless. I can't be any happier at the turn of events.
  - Hypothesis(**H**): I'm elated due to the latest development.
- High surface level match, but non-entailment.
  - Text(**T**): I am anything but a coffee addict.
  - Hypothesis(**H**): I am a coffee addict.
- Additional world knowledge requirement.
  - Text(**T**): Taj Mahal is located in Agra.
  - Hypothesis(**H**): Taj Mahal is located in India.

Outline

Technical Background

- **Excitement Open Platform (EOP)**:
  - Suite of open source state-of-the-art RTE algorithms.
  - Modular architecture supporting reuse of resources.

Outline

Previous Approaches - RTE

- Several classification and edit-distance based algorithms are available.
- Limitations:
  - Don't handle noisy data.
  - No use of context-based semantics.

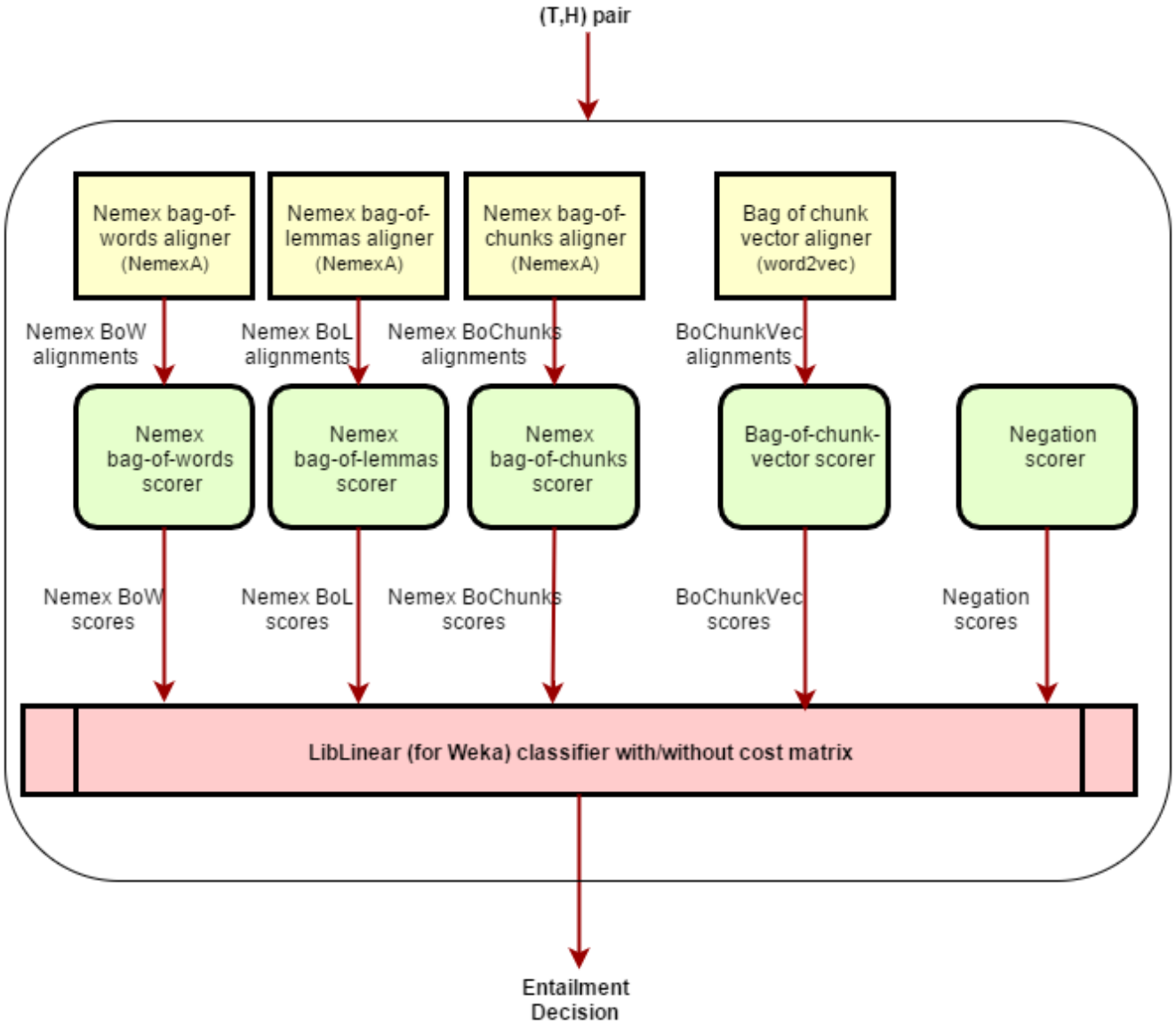
Outline

Textual Alignment

- What is text alignment?
  - Finding corresponding text snippets in T and H.
- Example:
  - Text(**T**): John Smith rode to Seattle and bought a Honda Civic.
  - Hypothesis(**H**): John drove to Seattle.

System architecture

EDA within Excitement Open Platform



Outline

Nemex Alignment

Approximate distance-based alignment

- Approximate alignment allows accounting for spelling differences and tokenization noise.
  - Text(**T**): Müller owns a dog.
  - Hypothesis(**H**): Mueller owned a pet.
- Exact match can be simulated from approximate match.

Nemex Alignment

Approximate distance-based alignment

- Approximate alignment allows accounting for spelling differences and tokenization noise.
  - Text(**T**): Müller owns a dog.
  - Hypothesis(**H**): Mueller owned a pet.
- Exact match can be simulated from approximate match.
- NemexA
  - Tool developed at DFKI to find approximately matching entries in a dictionary.
  - Supports n-gram vector-based match through cosine, Jaccard, Dice and overlap coefficients.
- Supported alignments: approximate bag-of-words, lemmas, chunks, and person name alignments.

Nemex Alignment

Pipeline

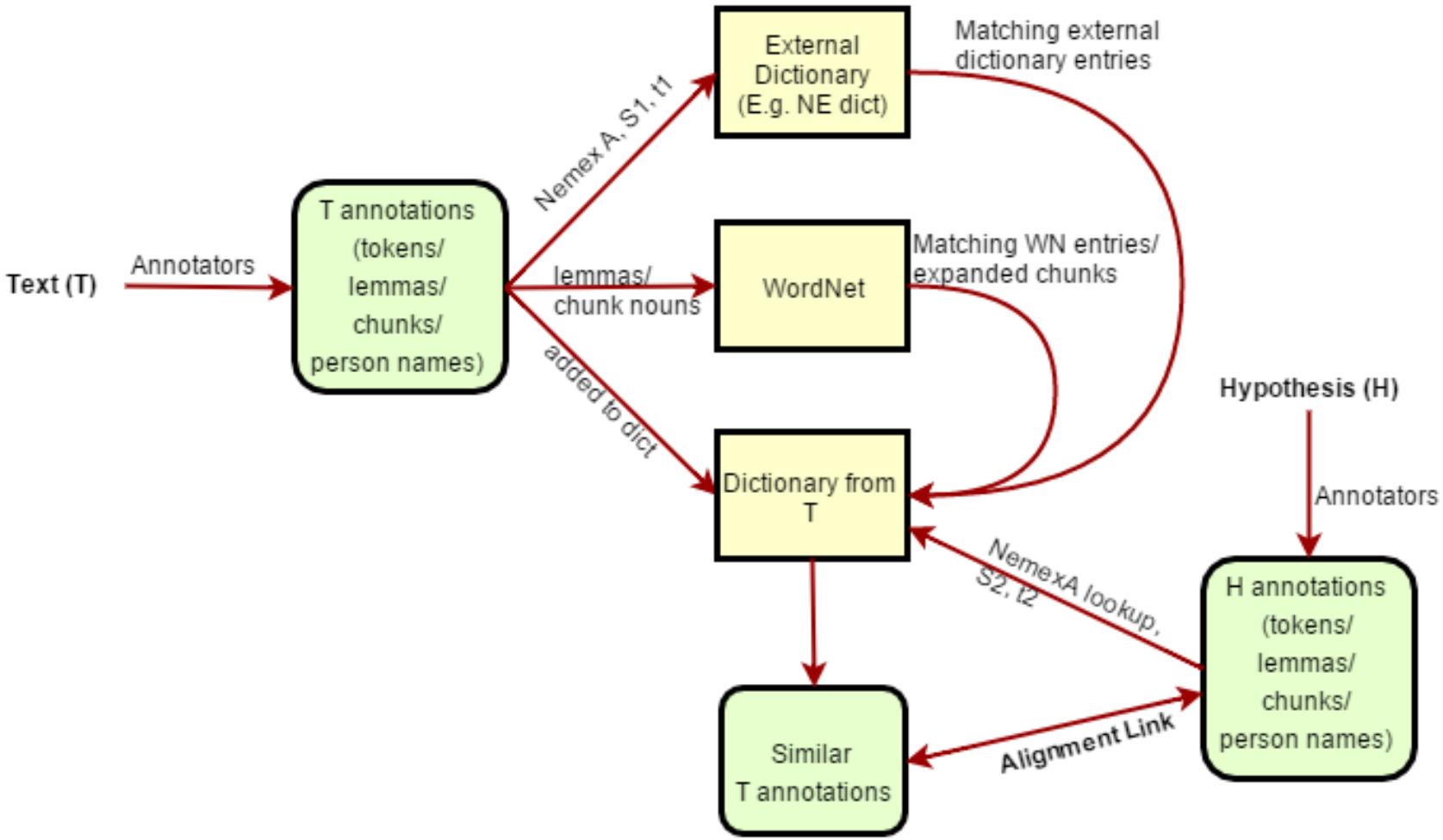


Figure: Nemex Aligners Pipeline - direction TtoH

Outline

Embedded Chunk Vectors Alignment

Using Word2Vec

- Embedded word vectors: High-dimensional distributed word vectors based on context.

Embedded Chunk Vectors Alignment

Using Word2Vec

- Embedded word vectors: High-dimensional distributed word vectors based on context.
- Why do we need it?
  - Approximate bag-of-words and lemmas match doesn't capture context.
  - Approximate bag-of-chunks match can't align chunk pairs like "hundreds of thousands" and "several thousands".
  - Embedded vectors capture both context and semantics.

Chunk Vector Alignment

Pipeline

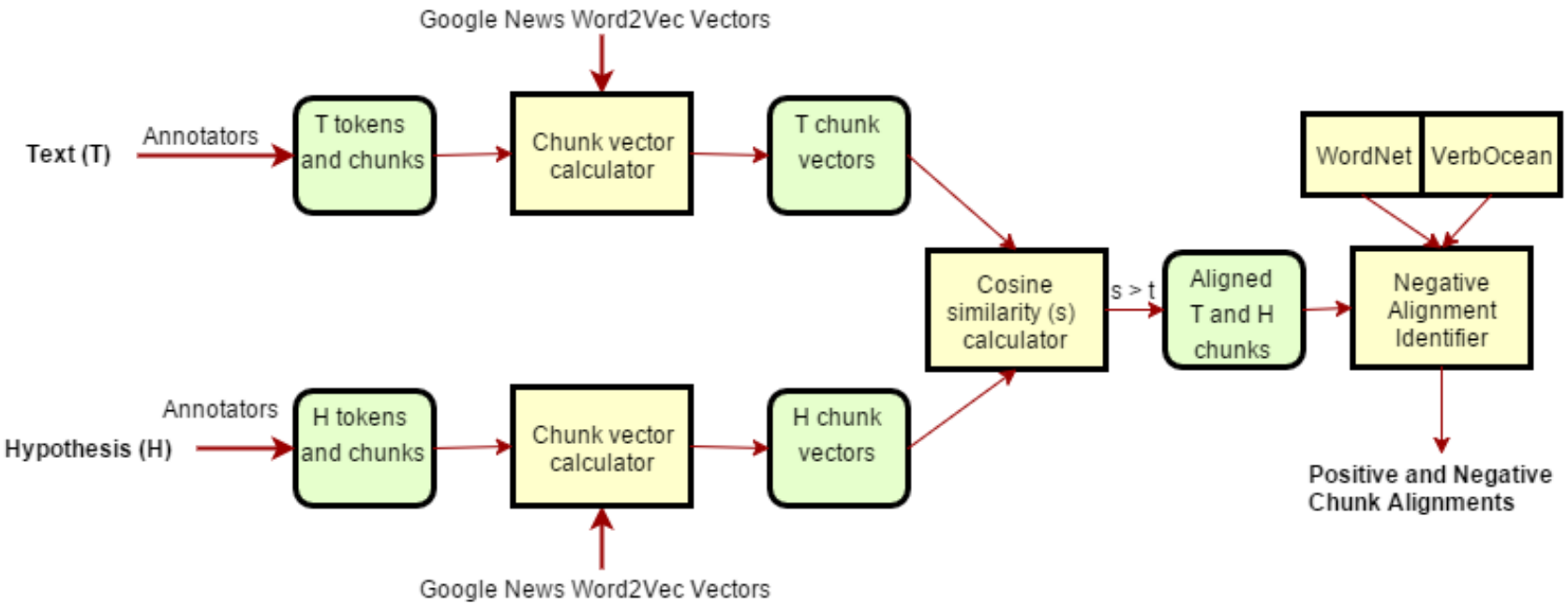


Figure: Embedded chunk vector aligner pipeline

Chunk Vectors Alignment

Positive and Negative Alignments

- Similar vectors: similar usage and meaning, not necessarily synonymous.
- Alignments may be negative w.r.t entailment. Example:
  - Text(**T**): He was a conservative politician.
  - Hypothesis(**H**): He was a liberal statesman.
- Within aligned chunks, negative alignments if:
  - Antonym token pair present in T and H: identified using WordNet.
  - H token is opposite of, stronger than or occurs before T token: identified using VerbOcean.

Outline

Nemex and Embedded vector-based Scorers

- Multiple scores calculated based on no. of alignments between T and H, relative to their lengths.
- Some score calculation depending on task the instance has been derived from.
- Multiple scores calculated based on coverage of words, nouns, verbs, adverbs and proper nouns under alignment.

Outline

Negation terms Scorer

- Negation words like “none”, “not”, etc. alter meaning of sentence significantly.
- Example:
  - Text(**T**): Spain did not win the championship.
  - Hypothesis(**H**): Spain won the championship.
  - Non-entailment
- Scores calculated through negation words frequency in T and H.

Outline

Classifier

- Binary classification, “entailing” and “non-entailing” classes.
- LibLinear algorithms for Weka used for classification.
- Logistic regression using L2 regularization applied after data randomization.

Outline

Data-set and Evaluation

- RTE3 data-set:
  - 800 (T,H) balanced pairs for training and development each.
  - Longer T - up to paragraph length, H is a single sentence.
  - Derived from IE, IR, Summarization and QA tasks.

Data-set and Evaluation

- RTE3 data-set:
  - 800 (T,H) balanced pairs for training and development each.
  - Longer T - up to paragraph length, H is a single sentence.
  - Derived from IE, IR, Summarization and QA tasks.
  - Results:

Proposed System	Best Alignment EDA	Best Previous EDA
65%	67%	66.8%

Table: Best accuracies on English data

- Proposed System: NemexBoW (0.8,cosine)+ BoChunkVec (0.8) + Negation
- Best Alignment EDA: Meteor alignments + coverage scores
- Best previous EDA: BIUTEE

Feature-Utility Analysis

Ablation tests and comparative results

Bag-of-Words	Approx Bag-of-words
63.25%	64.75%

Table: Baseline Accuracy: Exact BoW vs. Approximate BoW (0.8,cosine)

Bow+BoChunks	Bow+BoChunkVectors
62.875%	63.75%

Table: Accuracy: Approximate chunk vectors (0.7,cosine) vs. Embedded chunk vectors (0.7)

BoW+BoChunkVec	BoW+BoChunkVec+Neg
64.5%	64.75%

Table: Accuracy: Effect of Negation scoring (BoChunkVec:0.75)

Moving to bigger data-sets

- RTE3 data is very small; doesn't justify statistical approaches: need to move to larger data-sets.
- Ongoing evaluations with RTE6 and SNLI corpora.
- RTE-6:
  - Corpus based approach - find all sentences in corpus entailing H.
  - Candidate sentences retrieved from Lucene.
  - 15955 total pairs for training, highly unbalanced. 95% negative cases.
  - Indicates realistic entailment distribution.
- SNLI:
  - Very new corpus of 570k balanced (T,H) pairs for training, 10k pairs for development and testing each.
  - Makes nerual network training feasible.

Summary

- Approximate alignment can capture spelling variations and noise in data.
- Embedded vectors produce better chunk level alignments than surface level match.
- Negation terms scoring is useful, if such cases are present in data.
- Outlook
  - Statistical approaches work better on bigger data-sets.
  - Further semantic abstraction required for better inference.