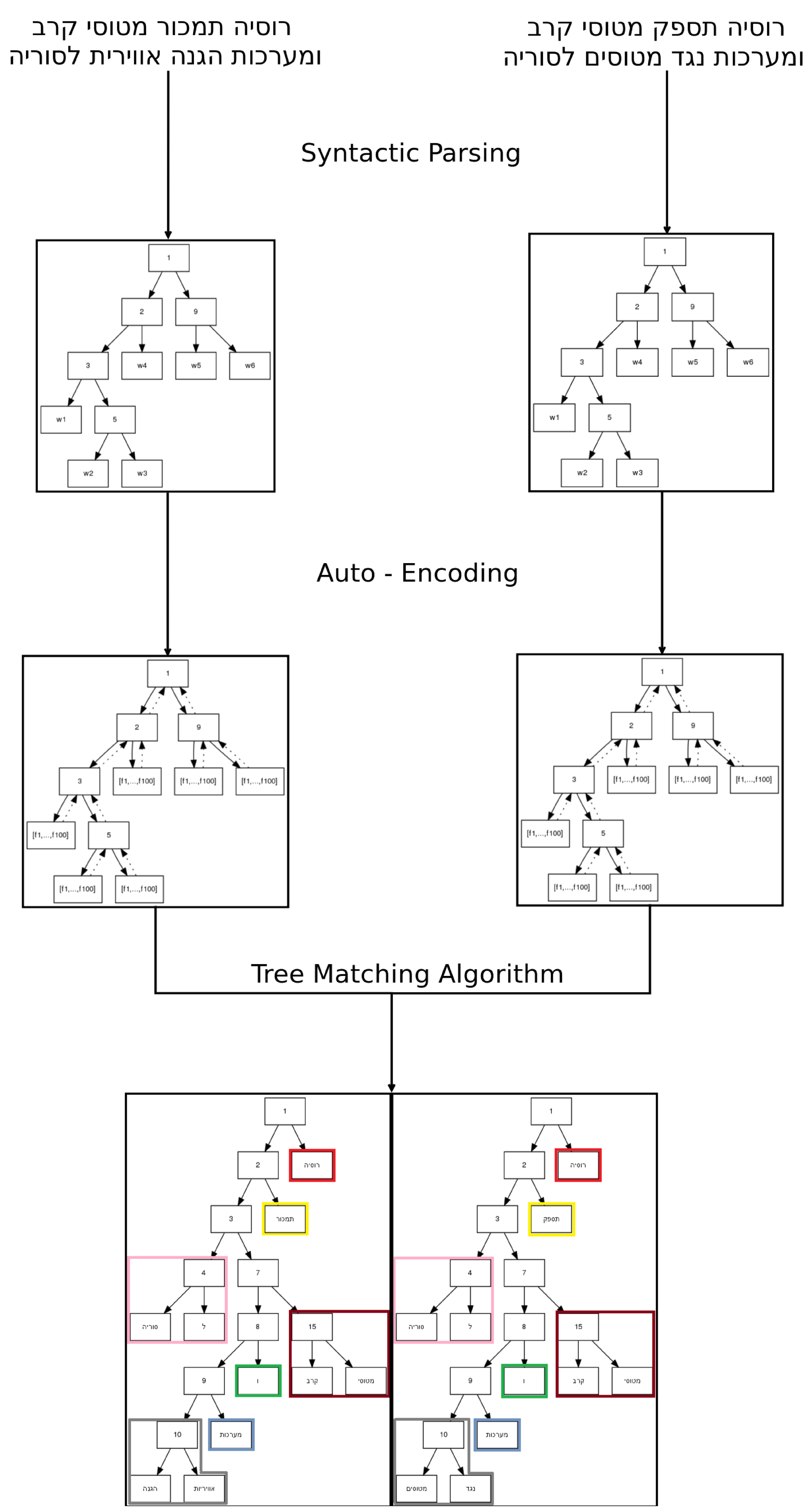
Hebrew Paraphrase Recognition Using Deep Learning Architecture

Gabriel Stanovsky, Advised By Prof. Michael Elhadad
Computer Science Department, Ben Gurion University

Background

- Text fragment A will entail text fragment B if a human being who trusts A, will consequently have to trust that B is also true.
- Text fragments (A, B) are said to be in Paraphrase Relationship if A entails B and vice-versa.
- Paraphrase Identification is the task of determining whether two given texts stand in a paraphrase relationship.
- Very little work has been done in the field for Hebrew.



Motivation

Automatic Summarization: Redundancy in the form of paraphrasing can be omitted in order to provide a shorter version of a document.

Automatic Filter of News Stream: Identification of the first story can be done by recognizing that a parallel news stream present paraphrases of the same story.

Automatic Construction of Thesaurus: Paraphrasing on the word level often introduce synonyms.Thus a Thesaurus can be assembled by aligning the changes between a paraphrase pair.

Syntactic Parsing

- A pre-trained model is used to obtain dependency and constituency parse trees.
- The parse trees are then binarized for the auto encoding process which follows.

Word Embedding

- A deep Learning architecture was trained to obtain a word embedding dictionary - a dictionary which maps between words and a 100 dimensional vector: $d(w) \in \mathbb{R}^{100}$
- These vectors were trained to maximize a specific language model.
- The leafs of the parse trees are replaced with word embedding using the aforementioned dictionary.

Auto - Encoding

- An auto-encoder model was trained to encode and decode between two 100 dimensional feature vector and one 100 dimensional feature vector: $D(E(r_1, r_2)) \approx (r_1, r_2)$
- This auto-encoder was trained to minimize the reconstruction error.
- By recursively applying the autoencoder - a feature vector is obtained also for internal nodes in the parse trees.

Tree Matching

- Given two auto-encoded parse trees t_1,t_2, define a Tree Match M, to be any set of tuples (n_1, n_2), where n_1, n_2 are nodes of t_1, t_2 accordingly, s.t. for every word w in $s_1(s_2)$, M contains exactly one tuple which contains a node in the path from w to the root of t_1 (t_2).
- This definition captures the idea that a paraphrase pair consists of sentences whose parts are interchangeable, from the sentence level down to the word level (including word-reordering).
- Following this definition, a score of a match can be defined:

$$S(M) = \sum_{(n_1, n_2) \in M} (||n_1, n_2||_2 \cdot (\text{\# spanned leaves by } n_1 \text{ and } n_2))$$

- We seek out to find a minimal match with regards to this metric. **This provides both an estimate of the probability of the pair being in a paraphrase relationship, as well as offering an alignment between the sentences**. This task was proven to be NP-complete.

Contribution and Experimental Results

Annotated Paraphrase Corpus:

- In order to test the framework, an Hebrew paraphrase corpus was collected.
- An algorithm was developed to acquire news articles from leading news sites, and align these based on the time they were published and their syntactic similarity.
- A very large unannotated corpus (about 1.4M headlines) of possible paraphrase pairs was collected.
- 1K of the possible pairs were tagged by human judges to obtain an annotated reference corpus for future research comparison.
- The proposed system was shown to achieve results compatible with the state of the art results for the English task:

Parse Type	Performance(ACC/F1)
Dependency	74.38 / 80.35
Constituency	69.20 / 74.83

Word Embedding:

- An embedding dictionary of 5K common Hebrew words was calculated, and proven to be useful as a plugin enhancer for supervised NLP tasks.
- The produced embeddings show improvement when adding them to a CRF POS tagger as additional features:

	without embeddings	with embeddings
TB1	0.879 / 0.735	0.900 / 0.804
A7	0.910 / 0.701	0.940 / 0.821
All	0.866 / 0.662	0.880 / 0.723

Performance on the English Corpus

- The proposed system was compared against the Microsoft Research Paraphrase corpus (MSRP) and achieved compatible results:

Measure	Performance	State of the Art
ACC	73.9 %	76.8 %
F1	82.4 %	83.6 %

Selected References:

Socher et al, 2011. Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection.
Ronan Collobert and Jason Weston, 2008. A unified architecture for natural language processing: deep neural networks with multitask learning.
Turian et al, 2010. Word representations: a simple and general method for semi-supervised learning.
Yoav Goldberg and Michael Elhadad, 2010. Easy-first dependency parsing of modern Hebrew.
Meni Adler, 2007. Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach.
Jordan B. Pollack, 1990. Recursive distributed representations.

Mail: gabriel.satanovsky@gmail.com http://www.cs.bgu.ac.il/~nlpproj/paraphrase/main.htm