Recognizing *textual entailment* (TE) is the task of deciding whether a sentence or a text implies another, e.g. the sentence ‘Ostriches put their heads into the sand to avoid the wind’ entails ‘Ostriches bury their heads in the sand’.

While a trivial task for humans in many ordinary situations, the problem of recognizing TE has proven extremely difficult for machine learning algorithms. Participants in the first *PASCAL* RTE workshop reported an accuracy of at most 59% (Dagan, Glickman, & Magnini, 2006).

Current approaches to the recognition of TE often use word alignment functions, exploiting syntactic and structural properties of the text. In order to find out whether semantic techniques are also valuable, we analyzed existing training sets from the *PASCAL* Challenges. Some semantic properties that are essential for defining the validity of an entailment were examined in detail and subsequently annotated.

The first stage consisted of analyzing semantic properties that are essential for TE. In the second stage, we annotated the most common relevant properties. In the third stage we revised the scheme and reapplied it on the same sentences. In the last stage we tested the scheme on new sentences.

We propose a widely applicable scheme for enriching training sets with semantic annotations in order to improve the performance of learning algorithms. Results indicate that the constructions of *restrictive modification*, e.g. an adjective (marked as *modifier* in figure 1) restricting a noun’s meaning (marked as *modifiee* in figure 1), and *apposition*, e.g. a phrase adding extra information to another (marked as *app* in figure 1), that occur in the text are often required for recognizing an entailment.
Figure 1. An annotated version of a sentence pair from the first Pascal challenge.

The concluded phrases from the upper sentence are labeled as rss in the lower sentence (called the hypothesis), each conclusion with the same id and sub-id as the phrase it refers to. The id relates each apposition or modifier construction with its counterpart in the hypothesis; the sub-id identifies the two parts of an apposition, which allows annotation of the part that is used in the hypothesis.

In general, a modifier construction is of the form ‘A B’, where ‘A’ restricts the meaning of ‘B’ and lead to conclusion ‘B’. In case of apposition the construction is ‘A, ...B..., C’, where ‘A’ and ‘B’ form the apposition, resulting in conclusion ‘A’ or ‘B’, or ‘A is B’.

We annotated a training set of 160 sentences while refining the annotation scheme, of which 56% contained at least one apposition or restrictive modifier. From the test set we used to test the applicability of the scheme on new data, we found 55% of the sentences containing at least one apposition or restrictive modifier. Further work include the semi-automatic annotation of a greater dataset and the testing of machine learning algorithms on the dataset.

Works Cited