# Sentence Pair Scoring: Towards Unified Framework for Text Comprehension

**Petr Baudiš**
FEE CTU Prague
Department of Cybernetics
Technická 2, Prague, Czech Republic
baudipet@fel.cvut.cz

**Jan Šedivý**
FEE CTU Prague
Department of Cybernetics
Technická 2, Prague, Czech Republic
sedivjan@fel.cvut.cz

## Abstract

We review the task of Sentence Pair Scoring, popular in the literature in various forms — slanted as Answer Sentence Selection, Paraphrasing, Semantic Text Scoring, Next Utterance Ranking, Recognizing Textual Entailment or e.g. a component of Memory Networks.

We argue that such tasks are similar from the model perspective (especially in the context of high-capacity deep neural models) and propose new baselines by comparing the performance of popular convolutional, recurrent and attention-based neural models across many Sentence Pair Scoring tasks and datasets. We discuss the problem of evaluating randomized models, propose a statistically grounded methodology, and attempt to improve comparisons by releasing new datasets that are much harder than some of the currently used well explored benchmarks.

To address the current research fragmentation in a future-proof way, we introduce a unified open source software framework with easily pluggable models, allowing easy evaluation on a wide range of semantic natural language tasks. This allows us to outline a path towards a universal machine learned semantic model for machine reading tasks. We support this plan by experiments that demonstrate reusability of models trained on different tasks, even across corpora of very different nature.

## 1 Introduction

A typical NLP machine learning task involves classifying a sequence of tokens such as a sentence or a document, i.e. approximating a function $f_1(s) \in [0, 1]$ (where $f_1$ may determine a domain, sentiment, etc.). But there is a large class of problems that involve classifying a pair of sentences, $f_2(s_0, s_1) \in \mathbb{R}$ (where $s_0$, $s_1$ are sequences of tokens, typically sentences).

Typically, the function $f_2$ represents some sort of *semantic similarity*, that is whether (or how much) the two sequences are semantically related. This formulation allows $f_2$ to be a measure for tasks as wide as topic relatedness, paraphrasing, degree of entailment, a pointwise ranking task for answer-bearing sentences or next utterance classification.

In this work, we adopt the working assumption that there exist certain universal $f_2$ type measures that may be successfuly applied to a wide variety of semantic similarity tasks — in the case of fully differentiable models, both architecture-wise and weight-wise, trained to represent universal semantic comprehension of sentences and adapted to the given task by just fine-tuning or adapting the final dense layer. Our argument for preferring $f_2$ to $f_1$ in this pursuit is the fact that the other sentence in the pair is essentially a very complex label when training the sequence model, which can therefore discern semantically rich structures and dependencies.

Determining and demonstrating such universal semantic comprehension models across multiple tasks remains a few steps ahead, since the research landscape is fragmented in this regard, with model research typically reported within the context of just a single $f_2$-type task, each dataset requiring sometimes substantial engineering work before measurements are possible, and results reported in ways that make meaningful model comparisons problematic. The primary purpose of this work is to unify research within a single framework that employs task-independent mod-

els, task-specific adaptation modules and unified statistically appropriate methodology for reportings. To demonstrate the feasibility of pursuing universal, task-independent $f_2$ models, we show that even simple neural models learn universal semantic comprehension as we improve their performance (even on relatively large datasets) by employing cross-task transfer learning.

The paper is structured as follows. In Sec. 2, we outline possible specific $f_2$ tasks and available datasets; in Sec. 3, we survey the popular basic baselines and the simpler currently employed neural models on these tasks; finally, in Sec. 4, we present model-task performances within a unified framework to establish the watermark for future research as well as gain insight into the suitability of models across a variety of tasks. In Sec. 5, we demonstrate that transfer learning across tasks is helpful to powerfully seed models. We conclude with Sec. 6, summarizing our findings and outlining several future research directions.

## 2 Tasks and Datasets

The tasks we are aware of that can be phrased as $f_2$-type problems are listed below. In general, we have decided to primarily focus on tasks that have reasonably large and realistically complex datasets freely available. On the contrary, we have explicitly avoided datasets that have licence restrictions on availability or commercial usage.

### 2.1 Answer Sentence Selection

Given a question and a set of candidate answer-bearing sentences, the task here is to rank higher sentences that are more likely to contain the answer to the question. As it is fundamentally an Information Retrival task in nature, the model performance is commonly evaluated in terms of Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR).

This task is popular in the NLP research community thanks to the dataset introduced in (Wang et al., 2007) which we refer to as `wang`, with six papers published between February 2015 and 2016 alone and neural models substantially improving over classical approaches based primarily on parse tree edits.[1] We can certainly call it the main research testbed for $f_2$-style task models. This task has also immediate applications e.g. in Question Answering systems.

In the context of practical applications, the so-far standard `wang` dataset has several downsides we observed when tuning and evaluating our models, illustrated numerically in Fig. 1 — the set of candidate sentences is often very small and quite uneven (which also makes rank-based measures unstable) and the total number of questions as well as individual sentence pairs is relatively small. Furthermore, the validation and test set are very small which makes for noisy performance measurements; the splits also seem quite different in the nature of questions since we see minimum correlation between performance on the validation and set tests, which calls the parameter tuning procedures and epoch selection for early stopping into question.

Alternative datasets WikiQA (Yang et al., 2015) and InsuranceQA (Tan et al., 2015) were proposed, but are encumbered by licence restrictions. Furthermore, we speculate that they may suffer from many of the problems above[2] (even if they are somewhat larger) and they did not gain much traction in the research community.

To alleviate the problems listed above, we are introducing a new dataset `yodaqa/curatedv2` based on the `curatedv2` question dataset (introduced in (Baudiš and Šedivý, 2015), further denoisified by Mechanical Turkers) with candidate sentences as retrieved by the YodaQA question answering system (Baudiš, 2015) from English Wikipedia. For models that can make substantial use of more data, we also include an even larger dataset `yodaqa/large2470` (though it has seen less gold standard denoisification), with its splits as supersets of the smaller splits.[3]

Fig. 1 compares the critical characteristics of the datasets. Furthermore, as apparent below, the baseline performances on the newly proposed datasets are much lower, which suggests that future improvements will be more apparent in evaluation.

[1] `http://aclweb.org/aclwiki/index.php?title=Question_Answering_(State_of_the_art)`

[2] For example, InsuranceQA is effectively a classification task rather than a ranking task, which we do not find as appealing in the context of practical applications.

[3] Note that the `wang` and `yodaqa` datasets however share a common ancestry regarding the set of questions and there may be some overlaps, even across train and test splits. Therefore, mixing training and evaluation on wang and yodaqa datasets within a single model instance is not advisable.

| Dataset | Train size | Val. size | Test size | Val.-Test $r$ | $\#s_0$ | $\#s_1$ per $s_0$ |
|---|---|---|---|---|---|---|
| wang | 44648 | 1149 | 1518 | -0.111 | 1492 | 34.9 $\pm$158% |
| yodaqa/curatedv2 | 69382 | 17406 | 82847 | 0.336 | 860 | 196.8 $\pm$98% |
| yodaqa/large2470 | 220846 | 55052 | 120069 | 0.546 | 2470 | 160.2 $\pm$97% |
| Ubuntu Dialogue v2 | 1M | 195600 | 189200 | | 38480 | 10 |

Figure 1: The Val.-Test column shows inter-trial Pearson's $r$ of validation and test MRRs, averaged across the models we benchmarked (see below). The last column includes relative standard deviation of the number of candidate sentences per question, which corresponds to the variation in the difficulty of the ranking task (as well as variation in expected measure values for individual questions). Ubuntu dataset shows $s_0$ and $s_1$ statistics just on val and test sets (training set are individual pairs).

### 2.2 Next Utterance Ranking

(Lowe et al., 2015) proposed a new large-scale and realistic dataset for an $f_2$-style task of ranking candidates for the next utterance in a chat dialog, given the dialog context. The technical formulation of the task is the same as for Answer Sentence Selection, but semantically, choosing the best followup involves different concerns than choosing an answer-bearing sentence.

The newly proposed Ubuntu Dialogue dataset is based on IRC chat logs of the Ubuntu community technical support channels and contains casually typed interactions regarding computer-related problems.[4] While the training set consists of individual labelled pairs of token sequences, evaluation is done by ranking 10 followups to given message(s) that are potentially relatively long (even more than 200 tokens).

Our primary motivation for using this dataset is its size. The numerical characteristics of this dataset are shown in Table 1.[5] We use the v2 version of the dataset.[6] Research published on this dataset so far relies on simple neural models. (Lowe et al., 2015) (Kadlec et al., 2015)

### 2.3 Semantic Textual Similarity

One of the canonical problems for $f_2$-type tasks is the STS track of the SemEval conferences (Agirrea et al., 2015). This is an annual competition of scoring pairs of sentences from 0 to 5 with the objective of maximizing correlation (Pearson's $r$) with manually annotated gold standard; the data is composed from diverse per-source splits that range from almost word-by-word paraphrases of newspaper titles to loosely related user forum questions. Every year, the past years are used as the training set.

Since 2016 results weren't released at the time of writing this paper, we use 2012 to 2014 as the training set, 2014 tweet-news split as the validation set, and 2015 splits (and the mean score across splits) as the testing set, making our results comparable to 2015 competition entrants. This means 3000 training pairs, 750 validation pairs and 8500 testing pairs. Contrary to Answer Sentence Selection, state-of-art methods are based on parse tree alignments (Sultan et al., 2015) and weren't beaten by neural models yet.

We also report results for this task on another dataset from the SemEval conferences, SICK-2014 (Marelli et al., 2014). In contrast to the STS track, it is geared at specifically benchmarking semantic compositional methods, aiming to capture only similarities on purely language and common knowledge level, without relying on domain knowledge, and there are no named entities or multi-word idioms. It consists of 4500 training pairs, 500 validation pairs and 4927 testing pairs.

### 2.4 Paraphrase Identification

Finally, a popular separately considered task is Paraphrase Identification, which can be thought of as a special case of Semantic Textual Scoring, but the task goal is binary classification rather than regression of a score on continuous scale.

The canonical dataset is the Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005), consisting of 4076 training and 1725 test pairs; 2/3 of the samples are labelled as 1 (is-a-paraphrase). State-of-art model uses an ensemble of hand-crafted overlap features (Ji and Eisenstein, 2013) and weren't beaten by neural models (Cheng and Kartsaklis, 2015) (He et al., 2015) yet.[7]

[4] In a manner, they resemble tweet data, but without the length description and with heavily technical jargon, command sequences etc.

[5] As in past papers, we use only the first 1M pairs (10%) of the training set.

[6] https://github.com/rkadlec/ubuntu-ranking-dataset-creator

[7] http://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_(State_of_the_art)

### 2.5 Other

Due to the very wide scope of the field, we leave some popular tasks and datasets as future work. In particular, this concerns the **Recognizing Textual Entailment** task (supported by the SNLI dataset) (Bowman et al., 2015) and the problem of memory selection in **Memory Networks** (supported by the baBi dataset) (Weston et al., 2015). A more realistic large paraphrasing dataset based on the AskUbuntu Stack Overflow forum had been recently proposed (Lei et al., 2015).

## 3 Models

As our goal is a universal text comprehension model, architecture-wise we focus on neural network models. We assume that the sequence is transformed using $N$-dimensional word embeddings on input, and employ models that produce a pair of sentence embeddings $E_0$, $E_1$ from the sequences of word embeddings $e_0$, $e_1$. Unless noted otherwise, a Siamese architecture is used that shares weights among both sentenes.

A scorer module that compares the $E_0, E_1$ sentence embeddings to produce a scalar result is connected to the model; for specific task-model configurations, we use either the **dot-product** module $E_0 \cdot E_1^T$ (representing non-normalized vector angle, as in e.g. (Yu et al., 2014) or (Weston et al., 2014)) or the **MLP** module that takes elementwise product and sum of the embeddings and feeds them to a two-layer perceptron with hidden layer of width $2N$ (as in e.g. (Tai et al., 2015)).[8] For the STS task, we follow this by score regression using class interpolation as in (Tai et al., 2015).

When training for a ranking task (Answer Sentence Selection), we use the bipartite ranking version of Ranknet (Burges et al., 2005) as the objective; when training for STS task, we use Pearson's $r$ formula as the objective; for binary classification tasks, we use the binary crossentropy objective.

### 3.1 Baselines

In order to anchor the reported performance, we report several basic methods. **Weighed word overlaps** metrics TF-IDF and BM25 (Robertson et al., 1995) are inspired by IR research and provide strong baselines for many tasks. We treat $s_0$ as the query and $s_1$ as the document, counting the number of common words and weighing them appropriately. IDF is determined on the training set.

The **avg** metric represents the baseline method when using word embeddings that proved successful e.g. in (Yu et al., 2014) or (Weston et al., 2014), simply taking the mean vector of the word embedding sequence and training an $U$ weight matrix $N \times 2N$ that projects both embeddings to the same vector space, $E_i = \tanh(U \cdot \bar{e_i})$, where the MLP scorer compares them. During training, $p = 1/3$ standard (elementwise) dropout is applied on the input embeddings.

A simple extension of the above are the **DAN** Deep Averaging Networks (Iyyer et al., 2015), which were shown to adequately replace much more complex models in some tasks. Two dense perceptron layers are stacked between the mean and projection, relu is used instead of tanh as the non-linearity, and word-level dropout is used instead of elementwise dropout.

### 3.2 Recurrent Neural Networks

**RNN** with memory units are popular models for processing sentenes (Tan et al., 2015) (Lowe et al., 2015) (Bowman et al., 2015). We use a bidirectional network with $2N$ GRU memory units[9] (Cho et al., 2014) in each direction; the final unit states are summed across the two per-direction GRUs to yield a $2N$ vector representation of the sentence. Like in the avg baseline, a projection matrix is applied on this representation and final vectors compared by an MLP scorer. We have found that applying massive dropout $p = 4/5$ both on the input and output of the network helps to avoid overfitting even early in the training.

### 3.3 Convolutional Neural Networks

**CNN** with sentence-wide pooling layer are also popular models for processing sentences (Yu et al., 2014) (Tan et al., 2015) (Severyn and Moschitti, 2015) (He et al., 2015) (Kadlec et al., 2015). We apply a multi-channel convolution (Kim, 2014) with single-token channel of $N$ convolutions and 2, 3, 4 and 5-token channels of $N/2$ convolutions each, relu transfer function, max-pooling over the whole sentence, and as above a projection to shared space and an MLP scorer. Dropout $p = 1/2$ is applied both on the input and output of

[8] The motivation is to capture both angle and euclid distance in multiple weighed sums. Past literature uses absolute difference rather than sum, but both performed equally in our experiments and we adopted sum for technical reasons.

[9] While the LSTM architecture is more popular, we have found the GRU results are equivalent while the number of parameters is reduced.

the convolution network.

### 3.4 RNN-CNN Model

The **RNN-CNN** model aims to combine both recurrent and convolutional networks by using the memory unit states in each token as the new representation of the token which is then fed to the convolutional network. Inspired by (Tan et al., 2015), the aim of this model is to allow the RNN to model long-term dependencies and model contextual representations of words, while taking advantage of the CNN and pooling operation for crisp selection of the gist of the sentence. We use the same parameters as for the individual models, except that we find applying dropout detrimental and need to reduce the number of parameters by using only $N$ memory units per direction.

### 3.5 Attention-Based Models

The idea of attention models is to attend preferrentially to some parts of the sentence when building its representation (Hermann et al., 2015) (Tan et al., 2015) (dos Santos et al., 2016) (Rocktäschel et al., 2015). There are many ways to model attention, as the initial proof of concept we adopt the (Tan et al., 2015) model **attn1511** which is conceptually simple and easy to implement. It asymmetrically extends the RNN-CNN model by extra links from $s_0$ CNN output to the post-recurrent representation of each $s_1$ token, determining an attention level for each token by weighed sum of the token vectors, and focusing on the relevant $s_1$ segment by transforming the attention levels using softmax and multiplying the token representations by the attention levels before they are fed to the convolutional network.

Convolutional network weights are not shared between the two sentences and the convolutional network output is not projected before applying the MLP scorer. The CNN used here is single-channel with $2N$ convolution filters 3 tokens wide.

## 4 Model Performance

### 4.1 `dataset-sts` framework

To easily implement models, dataset loaders and task adapters in a modular fashion so that any model can be easily run on any $f_2$-type task, we have created a new software package `dataset-sts` that integrates a variety of datasets, a Python dataset adapter `PySTS` and a Python library for easy construction of deep neural NLP models for semantic sentence pair scoring `KeraSTS` that uses the Keras machine learning library (Chollet, 2015). The framework is available for other researchers as open source on GitHub.[10]

### 4.2 Experimental Setting

We use $N = 300$ dimensional GloVe embeddings matrix pretrained on Wikipedia 2014 + Gigaword 5 (Pennington et al., 2014) that we keep adaptable during training; words in the training set not included in the pretrained model are initialized by random vectors uniformly sampled from $[-0.25, +0.25]$ to match the embedding standard deviation.

Word overlap is an important feature in many $f_2$-type tasks (Yu et al., 2014) (Severyn and Moschitti, 2015), especially when the sentences may contain named entities, numeric or other data for which no embedding is available. As a workaround, ensemble of world overlap count and neural model score is typically used to produce the final score. We try to adopt a more flexible approach, extending the embedding of each input token by several extra dimensions carrying boolean flags — bigram overlap, unigram overlap (except stopwords and interpunction), and whether the token starts with a capital letter or is a number.

Particular hyperparameters are tuned primarily on the `yodaqa/curatedv2` dataset unless noted otherwise in the respective results table caption. We apply $10^{-4}$ $L_2$ regularization and use Adam optimization with standard parameters (Kingma and Ba, 2014). In the answer selection tasks, we train on 1/4 of the dataset in each epoch. After training, we use the epoch with best validation performance; sadly, we typically observe heavy overfitting as training progresses and rarely use a model from later than a couple of epochs.

### 4.3 Evaluation Methodology

We report model performance averaged across 16 training runs (with different seeds). A consideration we must emphasize is that randomness plays a large role in neural models both in terms of randomized weight initialization and stochastic dropout. For example, the typical methodology for reporting results on the `wang` dataset is to evaluate and report a single test run after tuning on the

[10] Link redacted. Src snapshot included in submission.

| Model | MSR acc | MSR F1 |
|---|---|---|
| (Ji and Eisenstein, 2013) | 0.804 | 0.859 |
| (He et al., 2015) | 0.786 | 0.847 |
| always $y = 1$ | 0.665 | 0.799 |
| TF-IDF | 0.695 | 0.811 |
| BM25 | 0.695 | 0.811 |
| avg | 0.704 | 0.803 |
| | $\pm 0.004$ | $\pm 0.005$ |
| DAN | 0.704 | 0.802 |
| | $\pm 0.005$ | $\pm 0.010$ |
| RNN | 0.687 | 0.786 |
| | $\pm 0.030$ | $\pm 0.030$ |
| CNN | 0.695 | 0.792 |
| | $\pm 0.016$ | $\pm 0.020$ |
| RNN-CNN | 0.702 | 0.812 |
| attn1511 | 0.708 | 0.804 |

Figure 5: Model accuracy and F-measure on the MSR Paraphrase dataset.

dev set,[11] but `wang` test MRR has empirical standard deviation of 0.025 across repeated runs of our attn1511 model, which is more than twice the gap between every two successive papers pushing the state-of-art on this dataset! See the $^*$-marked sample in Fig. 2 for a practical example of this phenomenon.

Furthermore, on more complex tasks (Answer Sentence Selection in particular, see Fig. 1) the validation set performance is not a great approximator for test set performance and a strategy like picking training run with best validation performance would lead just to overfitting on the validation set.

To allow comparison between models (and with future models), we therefore report also 95% confidence intervals for each model performance estimate, as determined from the empirical standard deviation using Student's t-distribution.[12]

### 4.4 Results

In Fig. 2 to 5, we show the cross-task performance of our models. We can observe an effect analogous to what has been described in (Kadlec et al., 2015) — when the dataset is smaller, CNN models are preferrable, while larger dataset allows RNN models to capture the text comprehension task better. IR baselines provide strong competition and finding new ways to ensemble them with models should prove beneficial in the future.[13] This is especially apparent in the new Answer Sentence Selection datasets that have very large number of sentence candidates per question. The attention mechanism also has the highest impact in this kind of Information Retrieval task.

While our models clearly yet lag behind the state-of-art on the paraphrasing and STS tasks, it establishes the new baseline on the Ubuntu Dialogue dataset and it is not possible to statistically determine its relation to state-of-art on the `wang` Answer Sentence Selection dataset.

*Note to readers: We apologize for the fact that the numbers listed here are not entirely final. The Ubuntu dataset shows single val samples rather than 16-way test results and some other measurements are with fewer than 16 runs and confidence intervals not included. We expect to catch up with the measurements in just a couple of days. and do not expect paper conclusions to be affected in any way.*

## 5 Model Reusability

To confirm the hypothesis that our models learn a generic task akin to some form of text comprehension, we tried to train a model on the large Ubuntu Dialogue dataset (Next Utterance Ranking task) and then transfer the weights and retrain the model instance on the smaller `curatedv2` dataset (Answer Sentence Selection task).

We used the RNN model for the experiment in a configuration with dot-product scorer (which works much better on the Ubuntu dataset). The configuration, when trained from scratch on `curatedv2`, achieves MRR $0.371 \pm 0.023$, while with Ubuntu Dialogue pre-training it achieves MRR $0.493 \pm 0.021$. This represents a jump of about $0.12$ points and the resulting performance is comparable to a much more sophisticated attention model that is trained exclusively on the `curatedv2` dataset.

During our experiments, we have noticed that it is important not to apply dropout during retraining if it wasn't applied during the original training. We have also tried freezing the weights of some layers, but this never yielded a significant

[11] Confirmed by personal communication with paper authors.

[12] Over larger number of samples, this estimate converges to the normal distribution confidence levels. Note that the confidence interval covers the range of the true expected performance, not performance individually measured samples.

[13] We have tried simple averaging of predictions (as per (Kadlec et al., 2015)), but the benefit was small and inconsistent.

| Model | wang MAP | wang MRR | y.c.v2 MAP | y.c.v2 MRR | l2470 MAP | l2470 MRR |
|---|---|---|---|---|---|---|
| (Tan et al., 2015) | 0.728 | 0.832 | | | | |
| (dos Santos et al., 2016) | 0.753 | 0.851 | | | | |
| TF-IDF | 0.578 | 0.709 | 0.243 | 0.338 | 0.267 | 0.363 |
| BM25 | 0.630 | 0.765 | 0.294 | 0.485 | 0.314 | 0.491 |
| avg | 0.607 | 0.690 | 0.230 | 0.329 | 0.263 | 0.362 |
| | ±0.006 | ±0.010 | ±0.002 | ±0.004 | ±0.002 | ±0.006 |
| DAN | 0.643 | 0.735 | 0.233 | 0.354 | 0.273 | 0.387 |
| | ±0.100 | ±0.009 | ±0.003 | ±0.010 | ±0.003 | ±0.008 |
| RNN | 0.649 | 0.743 | 0.229 | 0.342 | 0.262 | 0.381 |
| | ±0.011 | ±0.010 | ±0.006 | ±0.011 | ±0.003 | ±0.008 |
| CNN | 0.691 | 0.770 | 0.229 | 0.309 | | |
| | ±0.007 | ±0.010 | ±0.005 | ±0.010 | | |
| RNN-CNN | 0.717 | 0.798 | 0.238 | 0.345 | | |
| | ±0.007 | ±0.011 | ±0.008 | ±0.015 | | |
| attn1511 | 0.708 | 0.790 | 0.275 | 0.469 | 0.288 | 0.431 |
| | ±0.009 | ±0.013 | ±0.007 | ±0.014 | ±0.006 | ±0.018 |
| *attn1511 | 0.756 | 0.859 | | | | |

Figure 2: Model results on the Answer Sentence Selection task, as measured on the `wang`, `yodaqa/curatedv2` and `yodaqa/large2470` datasets.
* Demonstration of the problematic single-measurement result reporting in past literature — an outlier sample in our 16-trial attn1511 benchmark that would score as a state of art; in total, two outliers in the trial (12.5%) scored better than (Tan et al., 2015).

improvement.

## 6 Conclusion

We have unified a variety of tasks in a single scientific framework of sentence pair scoring, and demonstrated a platform for general modelling of this problem and aggregate benchmarking of these models across many datasets. Promising initial transfer learning results suggest that a quest for generic neural model capable of task-independent text comprehension is becoming a meaningful pursuit. The open source nature of our framework and the implementation choice of a popular and extensible deep learning library allows for high reusability of our research and easy extensions with further more advanced models.

### 6.1 Future Work

Due to a very wide breadth of the $f_2$-problem scope, we were not able to cover all major tasks at once. Developing adapters, models and benchmarks for the task of **Recognizing Textual Entailment** remains the major next step, as well as for the problem of **Hypothesis Evidencing** where we would like to study the task of producing a binary classification of a hypothesis sentence $s_0$ based on a number of memory sentences $s_1$. We are in the process of developing realistic, hard datasets based on real-world event yes/no questions and newspaper snippets, as well as solving school test exams based on textbook and encyclopedia snippets.

We also did not include several major classes of models in our initial evaluation. Most notably, this includes serial RNNs with attention as used e.g. for the RTE task (Rocktäschel et al., 2015), and the skip-thoughts method of sentence embedding. (Kiros et al., 2015)

We believe that the Ubuntu Dialogue Dataset results demonstrate that the time is ripe to push the research models further towards the real world by allowing for wider sentence variability and less explicit supervision. But in particular, we believe that new models should be developed and tested on tasks with long sentences and wide vocabulary. Pushing the models to their limits will allow better differentiation regarding how much semantic text comprehension they are able to exhibit, which in turn should improve reusability, but datasets with low baseline performance will also better emphasize relative performance difference of individual models and make statistical significance more attainable.

In terms of models, recent work in many NLP domains (dos Santos et al., 2016) (Cheng et al., 2016) (Kumar et al., 2015) clearly points towards

| Model | MRR | 1-2 R@1 | 1-10 R@1 | 1-10 R@2 | 1-10 R@5 |
|---|---|---|---|---|---|
| * TF-IDF | | 0.749 | 0.488 | 0.587 | 0.763 |
| * RNN | | 0.777 | 0.379 | 0.561 | 0.836 |
| * LSTM | | 0.869 | 0.552 | 0.721 | 0.924 |
| avg | 0.618 | 0.787 | 0.464 | 0.602 | 0.831 |
| DAN | 0.616 | 0.783 | 0.465 | 0.594 | 0.823 |
| RNN | **0.786** | **0.910** | **0.671** | **0.805** | **0.953** |
| CNN | 0.659 | 0.805 | 0.528 | 0.639 | 0.842 |
| attn1511 | 0.773 | 0.903 | 0.654 | 0.788 | 0.948 |

Figure 3: Model results on the Ubuntu Dialogue next utterance ranking task. Models use slightly specific configuration due to much bigger dataset (in terms of both samples and sentence lengths) — only 160 tokens are considered per input, no dropout is applied, RNN use $N$ memory units, projection matrix is only $N \times N$ and the dot-product scorer is used for comparison. The attn1511 model furthermore has only $N/2$ RNN memory units and $N/2$ CNN filters.
* Exact models from (Lowe et al., 2015) reran on the v2 version of the dataset (by personal communication with Ryan Lowe) — note that the results in (Lowe et al., 2015) and (Kadlec et al., 2015) are on v1 and not directly comparable.

| Model | ans.for. | ans.stud | belief | headline | images | mean | SICK2014 |
|---|---|---|---|---|---|---|---|
| DLS@CU-S1 | 0.739 | 0.773 | 0.749 | 0.825 | 0.864 | 0.802 | |
| ECNU run1 | | | | | | | 0.841 |
| (Tai et al., 2015) | | | | | | | 0.868 |
| TF-IDF | 0.607 | 0.676 | 0.622 | 0.725 | 0.714 | 0.669 | 0.478 |
| BM25 | 0.626 | 0.690 | 0.632 | 0.725 | 0.718 | 0.678 | 0.474 |
| avg | 0.403 | 0.654 | 0.512 | 0.670 | 0.676 | 0.583 | 0.621 |
| | ±0.027 | ±0.009 | ±0.039 | ±0.011 | ±0.013 | ±0.134 | ±0.022 |
| DAN | 0.476 | 0.687 | 0.534 | 0.697 | 0.707 | 0.620 | 0.649 |
| | ±0.020 | ±0.006 | ±0.035 | ±0.006 | ±0.008 | ±0.119 | ±0.020 |
| RNN | 0.384 | 0.608 | 0.575 | 0.606 | 0.623 | 0.559 | 0.686 |
| | ±0.049 | ±0.040 | ±0.047 | ±0.067 | ±0.044 | ±0.110 | |
| CNN | 0.495 | 0.658 | 0.667 | 0.689 | 0.727 | 0.647 | 0.750 |
| | ±0.006 | ±0.006 | ±0.006 | ±0.006 | ±0.004 | ±0.098 | |
| RNN-CNN | 0.523 | 0.699 | 0.676 | 0.717 | 0.734 | 0.670 | |
| | ±0.007 | ±0.005 | ±0.010 | ±0.007 | ±0.005 | ±0.094 | |

Figure 4: Model results (Pearson's $r$) on the STS task — individual splits of the STS 2015 competition, mean STS 2015 performance and SICK2014 test set performance.

various forms of attention modelling to remove the bottleneck of having to compress the full spectrum of semantics into a single vector of fixed dimensionality. Another promising approach might be giving the network more flexibility regarding the final representation, for example by allowing it to remember a set of "facts" derived from each sentence; related work has been done on end-to-end differentiable shift-reduce parsers with LSTM as stack cells (Dyer et al., 2015).

In this paper, we have shown the benefit of training a model on a single dataset and then applying it on another dataset. One open question is whether we could jointly train a model on multiple tasks simultaneously (even if they do not share some output layers), for example by task-by-task batch interleaving during training. Another way to improve training would be to include extra supervision similar to the token overlap features that we already employ; for example, in the new Answer Sentence Selection task datasets, we can explicitly mark the actual tokens representing the answer.

## Acknowledgments

This work was financially supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/084/OHK3/1T/13, and the Forecast Foundation. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures."

We'd like to thank Tomáš Tunys, Rudolf Kadlec, Ryan Lowe, Cicero Nogueira dos santos and Bowen Zhou for helpful discussions and their insights, and Silvestr Stanko and Jiří Nádvorník for their software contributions.

## References


Eneko Agirrea, Carmen Baneab, Claire Cardiec, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirrea, Weiwei Guof, Inigo Lopez-Gazpioa, Montse Maritxalara, Rada Mihalceab, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability.

Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the YodaQA system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 222–228. Springer.

Petr Baudiš. 2015. YodaQA: A Modular Question Answering System Pipeline. In *POSTER 2015 - 19th International Student Conference on Electrical Engineering*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM.

Jianpeng Cheng and Dimitri Kartsaklis. 2015. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. *arXiv preprint arXiv:1508.02354*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

Franois Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *CoRR*, abs/1505.08075.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *EMNLP*.

Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez i Villodre. 2015. Denoising bodies to titles: Retrieving similar questions with recurrent convolutional models. *CoRR*, abs/1512.05726.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR*, abs/1506.08909.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *SemEval-2014*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Stephen E Robertson, Steve Walker, Susan Jones, et al. 1995. Okapi at trec-3.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 148–153.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, pages 22–32.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632.