Joint Learning of Sentence Embeddings for Relevance and Entailment

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

Abstract

We consider the problem of Recognizing Textual Entailment within an Information Retrieval context, where we must simultaneously determine the relevancy as well as degree of entailment for individual pieces of evidence to determine a yes/no answer to a binary natural language question.

We compare several variants of neural networks for sentence embeddings in a setting of decision-making based on evidence of varying relevance. We propose a basic model to integrate evidence for entailment, show that joint training of the sentence embeddings to model relevance and entailment is feasible even with no explicit per-evidence supervision, and show the importance of evaluating strong baselines. We also demonstrate the benefit of carrying over text comprehension model trained on an unrelated task for our small datasets.

Our research is motivated primarily by a new open dataset we introduce, consisting of binary questions and news-based evidence snippets. We also apply the proposed relevance-entailment model on a similar task of ranking multiple-choice test answers, evaluating it on a preliminary dataset of school test questions as well as the standard MCTest dataset, where we improve the neural model state-of-art.

1 Introduction

Let us consider the goal of building machine reasoning systems based on knowledge from fulltext data like encyclopedic articles, scientific papers or news articles. Such machine reasoning systems, like humans researching a problem, must be able to recover evidence from large amounts of retrieved but mostly irrelevant information and judge the evidence to decide the answer to the question at hand.

A typical approach, used implicitly in information retrieval (and its extensions, like IR-based Question Answering systems (Baudiš, 2015)), is to determine evidence relevancy by a keyword overlap feature (like tf-idf or BM-25 (Robertson et al., 1995)) and prune the evidence by the relevancy score. On the other hand, textual entailment systems that seek to confirm hypotheses based on evidence (Dagan et al., 2006) (Marelli et al., 2014) (Bowman et al., 2015) are typically provided with only a single piece of evidence or only evidence pre-determined as relevant, and are often restricted to short and simple sentences without open-domain named entity occurences. In this work, we seek to fuse information retrieval and textual entaiment recognition by defining the **Hypothesis Evaluation** task as deciding the truth value of a hypothesis by integrating numerous pieces of evidence, not all of it equally relevant.

As a specific instance, we introduce the **Argus Yes/No Question Answering** task. The problem is, given a real-world event binary question like *Did Donald Trump announce he is running for president?* and numerous retrieved news article fragments as evidence, to determine the answer for the question. Our research is motivated by the Argus automatic reporting system for the Augur prediction market platform. (Baudis et al., 2016b) Therefore, we consider the question answering task within the constraints of a practical scenario that has limited available dataset and only minimum supervision. Hence, authentic news sentences are the evidence (with noise like segmentation errors, irrelevant participial phrases, etc.), and whereas we have gold standard for the correct answers, the model must do without explicit super-

Proceedings of the 1st Workshop on Representation Learning for NLP, pages 8–17,
Berlin, Germany, August 11th, 2016.

vision on which individual evidence snippets are relevant and what do they entail.

To this end, we introduce an open dataset of questions and newspaper evidence, and a neural model within the Sentence Pair Scoring framework (Baudiš et al., 2016a) that (A) learns sentence embeddings for the question and evidence, (B) the embeddings represent both relevance and entailment characteristics as linear classifier inputs, and (C) the model aggregates all available evidence to produce a binary signal as the answer, which is the only training supervision.

We also evaluate our model on a related task that concerns ranking answers of multiple-choice questions given a set of evidencing sentences. We consider the MCTest dataset and the AI2-8grade/CK12 dataset that we introduce below.

The paper is structured as follows. In Sec. 2, we formally outline the Argus question answering task, describe the question-evidence dataset, and describe the multiple-choice questions task and datasets. In Sec. 3, we briefly survey the related work on similar problems, whereas in Sec. 4 we propose our neural models for joint learning of sentence relevance and entailment. We present the results in Sec. 5 and conclude with a summary, model usage recommendations and future work directions in Sec. 6.

2 The Hypothesis Evaluation Task

Formally, the Hypothesis Evaluation task is to build a function $y_i = f_h(H_i)$, where $y_i \in [0, 1]$ is a binary label (*no* towards *yes*) and $H_i = (q_i, E_i)$ is a hypothesis instance in the form of question text q_i and a set of $E_i = \{e_{ij}\}$ evidence texts e_{ij} as extracted from an evidence-carrying corpus.

2.1 Argus Dataset

Our main aim is to propose a solution to the Argus Task, where the **Argus** system (Baudis, 2015) (Baudis et al., 2016b) is to automatically analyze and answer questions in the context of the **Augur** prediction market platform.[1] In a prediction market, users pose questions about future events whereas others bet on the *yes* or *no* answer, with the assumption that the bet price reflects the real probability of the event. At a specified moment (e.g. after the date of a to-be-predicted sports match), the correct answer is retroactively determined and the bets are paid off. At a larger volume of questions, determining the bet results may present a significant overhead for running of the market. This motivates the Argus system, which should partially automate this determination — deciding questions related to recent events based on open news sources.

To train a machine learning model for the f_h function, we have created a dataset of questions with gold labels, and produced sets of evidence texts from a variety of news paper using a pre-existing IR (information retrieval) component of the Argus system. We release this dataset openly.[2]

To pose a reproducible task for the IR component, the time domain of questions was restricted from September 1, 2014 to September 1, 2015, and topic domain was focused to politics, sports and the stock market. To build the question dataset, we have used several sources:

- We asked Amazon Mechanical Turk users to pose questions, together with a golden label and a news article reference. This seeded the dataset with initial, somewhat redundant 250 questions.
- We manually extended this dataset by derived questions with reversed polarity (to obtain an opposite answer).
- We extended the data with questions auto-generated from 26 templates, pertaining top sporting event winners and US senate or gubernatorial elections.

To build the evidence dataset, we used the **Syphon** preprocessing component (Baudis et al., 2016b) of the Argus implementation[3] to identify semantic roles of all question tokens and produce the search keywords if a role was assigned to each token. We then used the IR component to query a corpus of newspaper articles, and kept sentences that contained at least 2/3 of all the keywords. Our corpus of articles contained articles from The Guardian (all articles) and from the New York Times (Sports, Politics and Business sections). Furthermore, we scraped partial archive.org historical data out of 35 RSS feeds from CNN, Reuters, BBC International, CBS News, ABC News, c—net, Financial Times, Skynews and the Washington Post.

[1] `https://augur.net/`

[2] `https://github.com/brmson/dataset-sts` directory data/hypev/argus

[3] `https://github.com/AugurProject/argus`

	Train	Val.	Test
Original $\#q$	1829	303	295
Post-search $\#q$	1081	167	158
Average $\#m$ per q.	19.04	13.99	16.66

Figure 1: Characteristics of the Argus QA dataset.

For the final dataset, we kept only questions where at least a single evidence was found (i.e. we successfuly assigned a role to each token, found some news stories and found at least one sentence with 2/3 of question keywords within). The final size of the dataset is outlined in Fig. 1 and some examples are shown in Fig. 2.

2.2 AI2-8grade/CK12 Dataset

The **AI2 Elementary School Science Questions** (no-diagrams variant)[4] released by the Allen Institute cover 855 basic four-choice questions regarding high school science and follows up to the Allen AI Science Kaggle challenge.[5] The vocabulary includes scientific jargon and named entities, and many questions are not factoid, requiring real-world reasoning or thought experiments.

We have combined each answer with the respective question (by substituting the *wh*-word in the question by each answer) and retrieved evidence sentences for each hypothesis using Solr search in a collection of CK-12 "Concepts B" textbooks.[6] 525 questions attained any supporting evidence, examples are shown in Fig. 3.

We consider this dataset as *preliminary* since it was not reviewed by a human and many hypotheses are apparently unprovable by the evidence we have gathered (i.e. the theoretical top accuracy is much lower than 1.0). However, we released it to the public[7] and still included it in the comparison as these qualities reflect many realistic datasets of unknown qualities, so we find relative performances of models on such datasets instructive.

2.3 MCTest Dataset

The **Machine Comprehension Test** (Richardson et al., 2013) dataset has been introduced to provide a challenge for researchers to come up with models that approach human-level reading comprehension, and serve as a higher-level alternative to semantic parsing tasks that enforce a specific knowledge representation. The dataset consists of a set of 660 stories spanning multiple sentences, written in simple and clean language (but with less restricted vocabulary than e.g. the bAbI dataset (Weston et al., 2015)). Each story is accompanied by four questions and each of these lists four possible answers; the questions are tagged as based on just *one* in-story sentence, or requiring *multiple* sentence inference. We use an official extension of the dataset for RTE evaluation that again textually merges questions and answers.

The dataset is split in two parts, MC-160 and MC-500, based on provenance but similar in quality. We train all models on a joined training set.

The practical setting differs from the Argus task as the MCTest dataset contains relatively restricted vocabulary and well-formed sentences. Furthermore, the goal is to find the single key point in the story to focus on, while in the Argus setting we may have many pieces of evidence supporting an answer; another specific characteristics of MCTest is that it consists of stories where the ordering and proximity of evidence sentences matters.

3 Related Work

Our primary concern when integrating natural language query with textual evidence is to find sentence-level representations suitable both for relevance weighing and answer prediction.

Sentence-level representations in the retrieval + inference context have been popularly proposed within the Memory Network framework (Weston et al., 2014), but explored just in the form of averaged word embeddings; the task includes only very simple sentences and a small vocabulary. Much more realistic setting is introduced in the Answer Sentence Selection context (Wang et al., 2007) (Baudiš et al., 2016a), with state-of-art models using complex deep neural architectures with attention (dos Santos et al., 2016), but the selection task consists of only retrieval and no inference (answer prediction). A more indirect retrieval task regarding news summarization was investigated by (Cao et al., 2016).

In the entailment context, (Bowman et al., 2015) introduced a large dataset with single-evidence sentence pairs (Stanford Natural Language Inference, SNLI), but a larger vocabulary and slightly more complicated (but still conservatively formed)

[4] `http://allenai.org/data.html`

[5] `https://www.kaggle.com/c/the-allen-ai-science-challenge`

[6] We have also tried English Wikipedia, but the dataset is much harder.

[7] `https://github.com/brmson/dataset-sts` directory `data/hypev/ai2-8grade`

Will Andre Iguodala win NBA Finals MVP in 2015? Should Andre Iguodala have won the NBA Finals MVP award over LeBron James? 12.12am ET Andre Iguodala was named NBA Finals MVP, not LeBron.
Will Donald Trump run for President in 2016? Donald Trump released Immigration Reform that will make America Great Again last weekend — . . . his first, detailed position paper since announcing his campaign for the Republican nomination . . . for president. The Fix: A brief history of Donald Trump blaming everything on President Obama DONALD TRUMP FOR PRESIDENT OF PLUTO!

Figure 2: Example pairs in the Argus dataset.

***pedigree chart model* is used to show the pattern of traits that are passed from one generation to the next in a family?** A pedigree is a chart which shows the inheritance of a trait over several generations. Figure 51.14 In a pedigree, squares symbolize males, and circles represent females.
***energy pyramid model* is used to show the pattern of traits that are passed from one generation to the next in a family?** Energy is passed up a food chain or web from lower to higher trophic levels. Each step of the food chain in the energy pyramid is called a trophic level.

Figure 3: Example pairs in the AI2-8grade/CK12 dataset. Answer texts substituted to a question are shown in italics.

sentences. They also proposed baseline recurrent neural model for modeling sentence representations, while word-level attention based models are being studied more recently (Rocktäschel et al., 2015) (Cheng et al., 2016).

In the MCTest text comprehension challenge (Richardson et al., 2013), the leading models use complex engineered features ensembling multiple traditional semantic NLP approaches (Wang and McAllester, 2015). The best deep model so far (Yin et al., 2016) uses convolutional neural networks for sentence representations, and attention on multiple levels to pick evidencing sentences.

4 Neural Model

Our approach is to use a sequence of word embeddings to build sentence embeddings for each hypothesis and respective evidence, then use the sentence embeddings to estimate relevance and entailment of each evidence with regard to the respective hypothesis, and finally integrate the evidence to a single answer.

4.1 Sentence Embeddings

To produce sentence embeddings, we investigated the neural models proposed in the `data-set-sts` framework for deep learning of sentence pair scoring functions. (Baudiš et al., 2016a)

We refer the reader to (Baudiš et al., 2016a) and its references for detailed model descriptions. We evaluate an **RNN** model which uses bidirectionally summed GRU memory cells (Cho et al., 2014) and uses the final states as embeddings; a **CNN** model which uses sentence-max-pooled convolutional filters as embeddings (Kim, 2014); an **RNN-CNN** model which puts the CNN on top of per-token GRU outputs rather than the word embeddings (Tan et al., 2015); and an **attn1511** model inspired by (Tan et al., 2015) that integrates the RNN-CNN model with per-word attention to build hypothesis-specific evidence embeddings. We also report the baseline results of **avg** mean of word embeddings in the sentence with projection matrix and **DAN** Deep Averaging Network model that employs word-level dropout and adds multiple nonlinear transformations on top of the averaged embeddings (Iyyer et al., 2015).

The original **attn1511** model (Baudiš et al., 2016a) (as tuned for the Answer Sentence Selection task) used a softmax attention mechanism that would effectively select only a few key words of the evidence to focus on — for a hypothesis-evidence token t scalar attention score $a_{h,e}(t)$, the focus $s_{h,e}(t)$ is:

$$s_{h,e}(t) = \exp(a_{h,e}(t)) / \sum_{t'} \exp(a_{h,e}(t'))$$

A different focus mechanism exhibited better performance in the Hypothesis Evaluation task, mod-

elling per-token attention more independently:

$$s_{h,e}(t) = \sigma(a_{h,e}(t)) / \max_{t'} \sigma(a_{h,e}(t'))$$

We also use relu instead of tanh in the CNNs.

As model input, we use the standard GloVe embeddings (Pennington et al., 2014) extended with binary inputs denoting token type and overlap with token or bigram in the paired sentence, as described in (Baudiš et al., 2016a). However, we introduce two changes to the word embedding model — we use 50-dimensional embeddings instead of 300-dimensional, and rather than building an adaptable embedding matrix from the training set words preinitialized by GloVe, we use only the top 100 most frequent tokens in the adaptable embedding matrix and use fixed GloVe vectors for all other tokens (including tokens not found in the training set). In preliminary experiments, this improved generalization for highly vocabulary-rich tasks like Argus, while still allowing the high-frequency tokens (like interpunction or conjunctions) to learn semantic operator representations.

As an additional method for producing sentence embeddings, we consider the **Ubu. RNN** transfer learning method proposed by (Baudiš et al., 2016a) where an RNN model (as described above) is trained on the Ubuntu Dialogue task (Lowe et al., 2015).[8] The pretrained model weights are used to initialize an RNN model which is then fine-tuned on the Hypothesis Evaluation task. We use the same model as originally proposed (except the aforementioned vocabulary handling modification), with the dot-product scoring used for Ubuntu Dialogue training replaced by MLP point-scores described below.

4.2 Evidence Integration

Our main proposed schema for evidence integration is **Evidence Weighing**. From each pair of hypothesis and evidence embeddings,[9] we produce two $[0, 1]$ predictions using a pair of MLP point-scorers of `dataset-sts` (Baudiš et al., 2016a)[10] with sigmoid activation function. The predictions are interpreted as $C_i \in [0, 1]$ entailment (0 to 1 as *no* to *yes*) and relevance $R_i \in [0, 1]$. To integrate the predictions across multiple pieces of evidence, we propose a weighed average model:

$$y = \frac{\sum_i C_i R_i}{\sum_i R_i}$$

We do not have access to any explicit labels for the evidence, but we train the model end-to-end with just y labels and the formula for y is differentiable, carrying over the gradient to the sentence embedding model. This can be thought of as a simple passage-wide attention model.

As a baseline strategy, we also consider **Evidence Averaging**, where we simply produce a single scalar prediction per hypothesis-evidence pair (using the same strategy as above) and decide the hypothesis simply based on the mean prediction across available evidence.

Finally, following success reported in the Answer Sentence Selection task (Baudiš et al., 2016a), we consider a **BM25 Feature** combined with Evidence Averaging, where the MLP scorer that produces the pair scalar prediction as above takes an additional BM25 word overlap score input (Robertson et al., 1995) besides the elementwise embedding comparisons.

5 Results

5.1 Experimental Setup

We implement the differentiable model in the Keras framework (Chollet, 2015) and train the whole network from word embeddings to output evidence-integrated hypothesis label using the binary cross-entropy loss as an objective[11] and the Adam optimization algorithm (Kingma and Ba, 2014). We apply $\mathbb{L}_2 = 10^{-4}$ regularization and a $p = 1/3$ dropout.

Following the recommendation of (Baudiš et al., 2016a), we report expected test set question accuracy[12] as determined by average accuracy in 16 independent trainings and with 95% confidence intervals based on the Student's t-distribution.

[8] The Ubuntu Dialogue dataset consists of one million chat dialog contexts, learning to rank candidates for the next utterance in the dialog; the sentences are based on IRC chat logs of the Ubuntu community technical support channels and contain casually typed interactions regarding computer-related problems, resembling tweet data, but longer and with heavily technical jargon.

[9] We employ Siamese training, sharing the weights between hypothesis and evidence embedding models.

[10] From the elementwise product and sum of the embeddings, a linear classifier directly produces a prediction; contrary to the typical setup, we use no hidden layer.

[11] Unlike (Yin et al., 2016), we have found ranking-based loss functions ineffective for this task.

[12] In the MCTest and AI2-8grade/CK12 datasets, we test and rank four hypotheses per question, whereas in the Argus dataset, each hypothesis is a single question.

Model	train	val	test
avg	0.872 ±0.009	0.816 ±0.008	0.744 ±0.020
DAN	0.884 ±0.012	0.822 ±0.011	0.754 ±0.025
RNN	0.906 ±0.013	0.875 ±0.005	0.823 ±0.008
CNN	0.896 ±0.018	0.857 ±0.006	0.822 ±0.007
RNN-CNN	0.885 ±0.010	0.860 ±0.007	0.816 ±0.009
attn1511	0.935 ±0.021	0.877 ±0.008	0.816 ±0.008
Ubu. RNN	0.951 ±0.017	0.912 ±0.004	**0.852** ±0.008

Figure 4: Model accuracy on the Argus task, using the evidence weighing scheme.

Model	Mean Ev.	BM25 Feat.	Weighed
avg	0.746 ±0.051	0.770 ±0.011	0.744 ±0.020
RNN	0.822 ±0.015	0.828 ±0.015	0.823 ±0.008
attn1511	0.819 ±0.013	0.811 ±0.012	0.816 ±0.008
Ubu. RNN	0.847 ±0.009	0.831 ±0.018	0.852 ±0.008

Figure 5: Comparing the influence of the evidence integration schemes on the Argus test accuracy.

5.2 Evaluation

In Fig. 4, we report the model performance on the Argus task, showing that the Ubuntu Dialogue transfer RNN outperforms other proposed models by a large margin. However, a comparison of evidence integration approaches in Fig. 5 shows that evidence integration is not the major deciding factor and there are no staticially meaningful differences between the evaluated approaches. We measured high correlation between classification and relevance scores with Pearson's $r = 0.803$, showing that our model does not learn a separate evidence weighing function on this task.

In Fig. 6, we look at the model performance on the AI2-8grade/CK12 task, repeating the story of Ubuntu Dialogue transfer RNN dominating other models. However, on this task our proposed evidence weighing scheme improves over simpler approaches — but just on the best model, as shown in Fig. 7. On the other hand, the simplest averaging model benefits from at least BM25 information to select relevant evidence, apparently.

Model	train	val	test
avg	0.505 ±0.024	0.442 ±0.022	0.401 ±0.016
DAN	0.556 ±0.038	0.491 ±0.015	0.391 ±0.008
RNN	0.712 ±0.053	0.381 ±0.016	0.361 ±0.012
CNN	0.676 ±0.056	0.442 ±0.012	0.384 ±0.011
RNN-CNN	0.582 ±0.057	0.439 ±0.024	0.376 ±0.014
attn1511	0.725 ±0.069	0.384 ±0.012	0.358 ±0.015
Ubu. RNN	0.570 ±0.059	0.494 ±0.012	**0.441** ±0.011

Figure 6: Model (question-level) accuracy on the AI2-8grade/CK12 task, using the evidence weighing scheme.

Model	Mean Ev.	BM25 Feat.	Weighed
avg	0.366 ±0.010	**0.415** ±0.008	**0.401** ±0.016
CNN	0.385 ±0.020		0.384 ±0.011
Ubu. RNN	0.416 ±0.011	0.418 ±0.009	**0.441** ±0.011

Figure 7: Comparing the influence of the evidence integration schemes on the AI2-8grade/CK12 test accuracy.

For the MCTest dataset, Fig. 8 compares our proposed models with the current state-of-art ensemble of hand-crafted syntactic and frame-semantic features (Wang and McAllester, 2015), as well as past neural models from the literature, all using attention mechanisms — the Attentive Reader of (Hermann et al., 2015), Neural Reasoner of (Peng et al., 2015) and the HABCNN model family of (Yin et al., 2016).[13] We see that averaging-based models are surprisingly effective on this task, and in particular on the MC-500 dataset it can beat even the best so far reported model of HABCNN-TE. Our proposed transfer model is statistically equivalent to the best model on both datasets (furthermore, previous work did not include confidence intervals, even though their models should also be stochastically initialized).

As expected, our models did badly on the multiple-evidence class of questions — we made no attempt to model information flow across ad-

[13](Yin et al., 2016) also reports the results on the former models.

Model	joint all (train)	MC-160 one	MC-160 multi	MC-160 all	MC-500 one	MC-500 multi	MC-500 all
hand-crafted		0.842	0.678	0.753	0.721	0.679	0.699
Attn. Reader		0.481	0.447	0.463	0.444	0.395	0.419
Neur. Reasoner		0.484	0.468	0.476	0.457	0.456	0.456
HABCNN-TE		0.633	**0.629**	**0.631**	0.542	**0.517**	0.529
avg	0.577 ±0.009	0.653 ±0.027	0.471 ±0.020	0.556 ±0.012	0.587 ±0.018	0.506 ±0.010	**0.542** ±0.011
DAN	0.590 ±0.009	0.681 ±0.017	0.486 ±0.010	0.577 ±0.010	**0.636** ±0.013	0.496 ±0.007	**0.560** ±0.007
RNN	0.608 ±0.030	0.583 ±0.033	0.490 ±0.018	0.533 ±0.020	0.539 ±0.016	0.456 ±0.013	0.494 ±0.012
CNN	0.658 ±0.021	0.655 ±0.020	0.511 ±0.012	0.578 ±0.014	0.571 ±0.013	0.483 ±0.012	0.522 ±0.009
RNN-CNN	0.597 ±0.039	0.617 ±0.041	0.493 ±0.021	0.551 ±0.020	0.554 ±0.023	0.470 ±0.016	0.508 ±0.014
attn1511	0.687 ±0.061	0.611 ±0.052	0.485 ±0.025	0.544 ±0.033	0.571 ±0.036	0.454 ±0.011	0.507 ±0.021
Ubu. RNN	0.678 ±0.035	**0.736** ±0.033	0.503 ±0.016	**0.612** ±0.023	**0.641** ±0.017	0.452 ±0.017	**0.538** ±0.015
* Ubu. RNN		0.786	0.547	0.658	0.676	0.494	0.577

Figure 8: Model (question-level) accuracy on the test split of the MCTest task, using the evidence weighing scheme. The first column shows accuracy on a train split joined across both datasets.
* The model with top MC-500 test set result (across 16 runs) that convincingly dominates HABCNN-TE in the *one* and *all* classes and illustrates that the issue of reporting evaluation spread is not just theoretical. 5/16 of the models have MC-160 *all* accuracy > 0.631.

Model	Mean Ev.	BM25 Feat.	Weighed
avg	0.423 ±0.014	0.506 ±0.012	**0.542** ±0.011
CNN	0.373 ±0.036	**0.509** ±0.027	**0.522** ±0.009
Ubu. RNN	0.507 ±0.014	0.509 ±0.012	**0.538** ±0.015

Figure 9: Comparing the influence of the evidence integration schemes on the MC-500 (all-type) test accuracy.

jacent sentences in our models as this aspect is unique to MCTest in the context of our work.

Interestingly, evidence weighing does play an important role on the MCTest task as shown in Fig. 9, significantly boosting model accuracy. This confirms that a mechanism to allocate attention to different sentences is indeed crucial for this task.

5.3 Analysis

While we can universally proclaim Ubu. RNN as the best model, we observe many aspects of the Hypothesis Evaluation problem that are shared by the AI2-8grade/CK12 and MCTest tasks, but not by the Argus task.

Our largest surprise lies in the ineffectivity of evidence weighing on the Argus task, since observations of irrelevant passages initially led us to investigate this model. We may also see that non-pretrained RNN does very well on the Argus task while CNN is a better model otherwise.

An aspect that could explain this rift is that the latter two tasks are primarily *retrieval* based, where we seek to judge each evidence as irrelevant or essentially a paraphrase of the hypothesis. On the other hand, the Argus task is highly *semantic* and compositional, with the questions often differing just by a presence of negation — recurrent model that can capture long-term dependencies and alter sentence representations based on the presence of negation may represent an essential improvement over an n-gram-like convolutional scheme. We might also attribute the lack of success of evidence weighing in the Argus task to a more conservative scheme of passage retrieval employed in the IR pipeline that produced the dataset. Given the large vocabulary and noise levels in the data, we may also simply require more data to train the evidence weighing properly.

We see from the training vs. test accuracies that RNN-based models (including the word-level attention model) have a strong tendency to overfit on our small datasets, while CNN is much more resilient. While word-level attention seems appealing for such a task, we speculate that we simply might not have enough training data to properly train it.[14] Investigating attention transfer is a point for future work — by our preliminary experiments on multiple datasets, attention models appear more task specific than the basic text comprehension models of memory based RNNs.

One concrete limitation of our models in case of the Argus task is a problem of reconciling particular named entity instances. The more obvious form of this issue is *Had Roger Federer beat Martin Cilic in US OPEN 2014?* versus an opposite *Had Martin Cilic beat Roger Federer in US OPEN 2014?* — another form of this problem is reconciling a hypothesis like *Will the Royals win the World Series?* with evidence *Giants Win World Series With Game 7 Victory Over Royals*. An abstract embedding of the sentence will not carry over the required information — it is important to explicitly pass and reconcile the roles of multiple named entities which cannot be meaningfully embedded in a GloVe-like semantic vector space.

6 Conclusion

We have established a general Hypothesis Evaluation task with three datasets of various properties, and shown that neural models can exhibit strong performance (with less hand-crafting effort than non-neural classifiers). We propose an evidence weighing model that is never harmful and improves performance on some tasks. We also demonstrate that simple models can outperform or closely match performance of complex architectures; all the models we consider are task-independent and were successfully used in different contexts than Hypothesis Evaluation (Baudiš et al., 2016a). Our results empirically show that a basic RNN text comprehension model well trained on a large dataset (even if the task is unrelated and vocabulary characteristics are very different) outperforms or matches more complex architectures trained only on the dataset of the task at hand.[15]

Finally, on the MCTest dataset, our best proposed model is better or statistically indistinguishable from the best neural model reported so far (Yin et al., 2016), even though it has a simpler architecture and only a naive attention mechanism.

We would like to draw several recommendations for future research from our findings: (A) encourage usage of basic neural architectures as evaluation baselines; (B) suggest that future research includes models pretrained on large data as baselines; (C) validate complex architectures on tasks with large datasets if they cannot beat baselines on small datasets; and (D) for randomized machine comprehension models (e.g. neural networks with random weight initialization, batch shuffling or probabilistic dropout), report expected test set performance based on multiple independent training runs.

As a general advice for solving complex tasks with small datasets, besides the point (B) above our analysis suggests convolutional networks as the best models regarding the tendency to overfit, unless semantic composionality plays a crucial role in the task; in this scenario, simple averaging-based models are a great start as well. Preinitializing a model also helps against overfitting.

We release our implementation of the Argus task, evidence integration models and processing of all the evaluated datasets as open source.[16]

We believe the next step towards machine comprehension NLP models (based on deep learning but capable of dealing with real-world, large-vocabulary data) will involve research into a better way to deal with entities without available embeddings. When distinguishing specific entities, simple word-level attention mechanisms will not do. A promising approach could extend the flexibility of the final sentence representation, moving from attention mechanism to a memory mechanism[17] by allowing the network to remember a set of "facts" derived from each sentence; related work has been done for example on end-to-end differentiable shift-reduce parsers with LSTM as stack cells (Dyer et al., 2015).

Acknowledgments

This work was co-funded by the Augur Project of the Forecast Foundation and financially supported by the Grant

[14] Just reducing the dimensionality of hidden representations did not yield an improvement.

[15] Even if these use multi-task learning, which was employed in case of the HABCNN models that were trained to also predict question classes.

[16] `https://github.com/brmson/dataset-sts` task *hypev*

[17] Not necessarily "memories" in the sense of Memory Networks.

Agency of the Czech Technical University in Prague, grant No. SGS16/ 084/OHK3/1T/13. 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 Peronet Despeignes of the Augur Project for his support. Carl Burke has provided instructions for searching CK-12 ebooks within the Kaggle challenge.

References

Petr Baudiš, Jan Pichl, Tomáš Vyskočil, and Jan Sedivý. 2016a. Sentence pair scoring: Towards unified framework for text comprehension. *CoRR*, abs/1603.06127.

Petr Baudis, Silvestr Stanko, and Peronet Despeignes. 2016b. Argus: An artificial-intelligence assistant for augur's prediction market platform reporters.

Petr Baudis. 2015. Argus: Deciding questions about events.

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 EMNLP Conference*. ACL.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2016. Attsum: Joint learning of focusing and summarization with neural attention. *arXiv preprint arXiv:1604.00125*.

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`.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

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.

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.

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 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*.

Baolin Peng, Zhengdong Lu, Hang Li, and Kam-Fai Wong. 2015. Towards neural network-based reasoning. *arXiv preprint arXiv:1508.05508*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the EMNLP 2014*, 12:1532–1543.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text.

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.

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

Hai Wang and Mohit Bansal Kevin Gimpel David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. *Proceedings of ACL*, Volume 2: Short Papers:700.

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.

Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. 2016. Attention-based convolutional neural network for machine comprehension. *CoRR*, abs/1602.04341.