Understanding and Improving Neural Models for Natural Language Interference

Verständnis und Verbesserung Neuronaler Modelle für Natural languge inference
Master-Thesis von Max Glockner
April 2018

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Physik
Institut f"ur Angewandte Festkernphysik
Speerspitze der Elite

Understanding and Improving Neural Models for Natural Language Interference
Verständnis und Verbesserung Neuronaler Modelle für Natural languge inference

Vorgelegte Master-Thesis von Max Glockner

1. Gutachten: Prof. Dr. Iryna Gurevych
2. Gutachten: Dr. Yoav Goldberg
3. Gutachten: Dr. Andreas Rücké

Tag der Einreichung:

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-12345
URL: http://tuprints.ulb.tu-darmstadt.de/1234

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Erklärung zur Abschlussarbeit gemäß §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, J. Walker, die vorliegende Master-Thesis ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Datum / Date:	Unterschrift / Signature:
______________________	______________________________________

Abstract

TODO in English...

Zusammenfassung

TODO in Deutsch...

List of abbreviations

biLSTM bidirectional Long-Short-Term Memory Network

BoW Bag of Words

ESIM Enhanced Sequential Inference Model

IE Information Extraction

IR Information Retrieval

KIM Knowledge-based Inference Model

LSTM Long-Short-Term-Memory

MLP Multi Layer Perceptron

MultiNLI MultiGenre Natural Language Inference Corpus

NLI Natural Language Inference

NLP Natural Language Processing

NLU Natural Language Understanding

POS Part of Speech

OANC Open American National Corpus

QA Question Answering

RNN Recurrent Neural Network

RTE Recognizing Textual Entailment

SD Standard Deviation

SICK Sentences Involving Compositional Knowledge

SNLI The Stanford Natural Language Inference Corpus

WSD Word Sense Disambiguation

YAGO Yet Another Great Ontology

Contents

1 Introduction

In recent years neural networks again gained a lot of popularity for many machine learning tasks, including the field of Natural Language Processing (NLP). While previous generation solutions heavily depended on handcrafted features, these models are capable of learning meaningful feature representations automatically (Bengio et al., 2013), thus avoiding time consuming process of feature-engineering. For the most part, neural networks solely rely on distributed word representations, also known refered to as word-embeddings, like word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) and typically learn fixed-length dense vector representations for the input text. While they provie strong generalization capabilities, they fail to capture simple world-knowledge (Celikyilmaz et al., 2010) and even have trouble differentiationg between mutually exclusive words, if they generally are used in similar contexts (Vulić et al., 2017). As opposed to that, traditional approaches execcively made usage of lexical resources containing relational and factual information about words and entities, thus providing a huge amount of ready-to-sue knowledge bases. Intuitivly, combining both worlds by integrating existing knowldge bases into neural networks should even further improve these models, due to a more sophisticated Natural Language Understanding (NLU). This is analogue tothe way humans understand text, by having a solid understanding of the world, that influences the subjective interpretation of every word within a sentence. Given the sentence "The official language in the USA is English." an average human can conclude that the official language of *New York* also is English, knowing that *New York* is within the *USA*.

To improve the NLU of neural models for NLP we address the mentioned problems by analysing the sentence representations of a state-of-the-art model and identifying knowledge that is captured or not captured using state-of-the-art strategies without external resources. We show those state-of-the-art models are limited in their generalization ability and fail to capture simple inferences. To overcome this problem, we evaluate how additional knowledge from exxternal resources could be inferred neural networks. While our aim is to provide generally applicable results, we base our experiments on the task of Natural Language Inference (NLI) (Bowman et al., 2015), also known as Recognizing Textual Entailment (RTE) (Dagan et al., 2006). As this is known to be a fundamental task for NLU (MacCartney and Manning, 2007), insights gained here can improve other tasks of NLP that indirectly depend on it.

Structure of the Thesis

While we explain relevant techniques and concepts, we expect the reader to have a basic understanding of common machine-learning practices, neural networks, including basic network architectures like Long-Short-Term-Memory (LSTM) or Recurrent Neural Network (RNN), and NLP in general. This thesis is structured in the following manner:

- Section §2 is used to give definitions for NLI and relevant word-relations. We further give a detailed description about the architecture and traning of the state-of-the-art model, that we use throughout all our experiments.
- In section §3 we introduce recently publish relevant datasets for NLIand discuss several strategies proved to be successful. In addition we show a selectionn of lexical resources that contain relevant information to improve the NLU of neural models and various strategies that have been applied to integrate them.
- We analyse how the information of a natural language text is encoded within the sentence representation of a neural model and give insights on how the model uses it in section §4.
- We derive a new testset from a major dataset for NLI, demonstrating the poor generalization abilities of state-of-the-art models in section §5.
- Based on the new data we evaluate whether external resources are helpful for the task using advanced embeddings and multitask-learning.

2 Theoretical Background

This section gives an overview of the task that is used within this work (§2.1) and relevant datasets regarding this task (§3.2).

2.1 Natural Language Inference

NLI (Bowman et al., 2015) deals with the problem to identify, whether one piece of natural text, namely the *hypothesis*, can be inferred from another piece of text, namely the *premise*. The hypothesis h is said to be entailed by the premise p if a human reader would conclude that the hypothesis is true, given the fact that the premise is true. Therefore it differes from strict logical inference. While in NLI a high plausability for the premise to imply the hyothesis, based on the human judgement, is sufficient, the latter one strives to achieve certainity (**?**). NLI essentially breaks down to an alignment problem (MacCartney et al., 2008). Given the sentence pair

Premise:	*Donald Trump is eating his cheeseburger in his bedroom.*
Hypothesis:	*The president of the United States is snacking a cheeseburger in the White House.*

the model is required to correctly align *Donald Trump* with *The president of the United States*, *eating* with *snacking* and have information that his *bedroom* is within the *White House*. Here it can be seen, how the system would not only need to cope with different ways of expressing the same meaning, due to the nature of language, but also is required to access and process factual information, that is commonly known to an average human.

Relatedness to other NLP tasks

While NLI clearly is central to reasoning capabilities, it is very fundamental and applicable to a large variety of NLP as the ability to recognize textual entailment is a fundamental and necessary problem towards real NLU (MacCartney and Manning, 2007; Bos and Markert, 2005). Many NLP applications such as Question Answering (QA), Summarization Information Extraction (IE) implicitly depend on this ability, as the huge variability of possible expressions for the same meaning it is a core phenomen of natural language (**?**). All three tasks require the model to infer that the target meaning of interest can be inferred from any other variant of textual expression. For QA it is the identification of a correct answer, for summarization, the complete summary needs to be implied by the original text and similarily, while redundant sentences expressing the same meaning should be ommited. Similarly IE, especially if using multiple documents, needs to infer, whether two variants of text contain the same information. Even simple paraphrasing can be broken down to a lexical inference problem with mutual entailment between p and h. As end applications for NLP in addition to NLU need to solve another complicated machine-learning task it is hard to compare and directly improve their NLU capabilities. Thus, one of the main purposes of NLI, being a very basic problem towards NLU, serves a a benchmark with any improvements helping a large variety of high level tasks (Williams et al., 2017; Cooper et al., 1996; Bos and Markert, 2005; Dagan et al., 2006).

2.2 Lexical Semantic Relations

Lexical relations describe the relationship between words[1] whereas *Lexical Semantic Relations* specifically indicate relations referring to the meaning of the word. (Murphy, 2003) and have shown to be helpful for detecting lexical inferences (**?**). We define the following relations based on those of Jurafsky and Martin (2008). One key characteristic of natural language is ambiguity, tht is also present in lexical semantics as words may have several meanings or *senses*[2]. To deal with this phenomene, lexical semantic relations are defined between senses rather than words. For the sake of simplicity for the most part we follow a naive approach in the following chapters of assuming the most dominant sense of a word, when referring to it. Specifically we define *Synonymy*, *Antonomy*, *Hypernomy* and *Holonomy*, the latter two relations are visualized[3] in Figure 1.

2.2.1 Synonymy and antonomy

Synonomy is a symmetric relationship between two senses or two words. Two senses of two different words are said to be synonyms, if they have the same or nearly the same meaning. Synonymy between words is holds, if one word can

[1] While there is no single definition for *word*, we refer assume a single word to have the same surface form and the same lemma.

[2] The phenomen of words having multiple senses is called *homonymy*, if both senses show indicate no relation but share the surface form like "bank" (financial institution) and "bank" (sloping mound). If those senses are semantically related like "milk" (take milk from female mammals) and "milk" (like cow's milk), the relationship is called *polysemy* (Jurafsky and Martin, 2008)

[3] This is for illustration purposes only and we only added some relevant relations between the entities, more relation are possible. For instance, the holonym relationship would of course hold between *head* and any other *animal*.

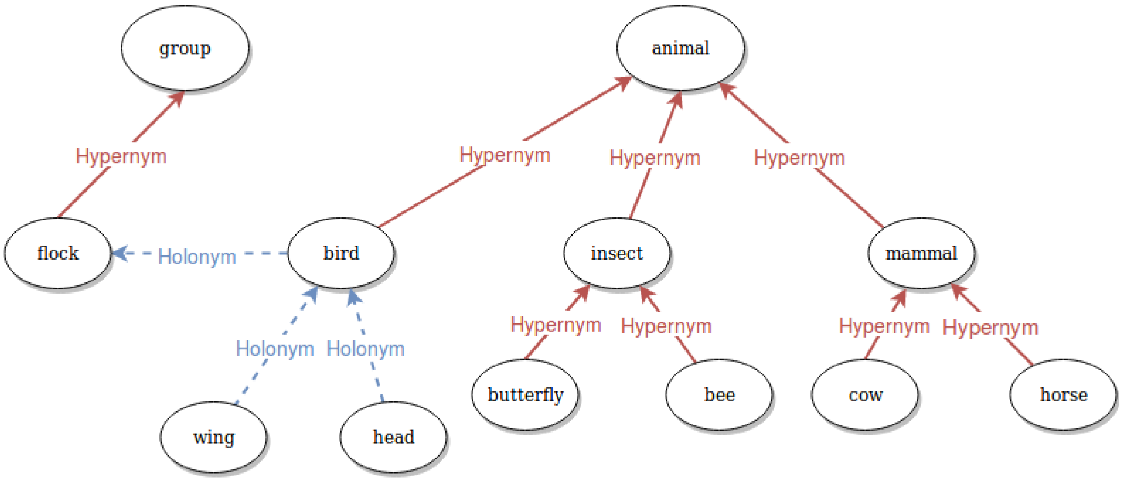

Figure 1: A sample ontology of animals to illustrate the lexical relations *Hypernomy* and *Holonymy*.

be replaced by the other word in any sentence without changing the meaning of the sentence. True synonyms are rare, as most words at least have subtle differences in their meaning or are used within different contexts. We thus follow common practice and loosen the strict definition by refering to synonmys if they have approximately similar meanings. Like synonomy, antomy is a symmetric relationship between senses, however having the opposite meaning, which might be caused by a binary opposition like "opened/closed", by different ends on some scale like "hot/cold" or by directional change like "upwards/downwards". Since antonyms semantically are identical in all other aspects with synonyms, these relations are hard to distinguish from each other automatically.

2.2.2 Hypernomy

Hypernomy (or Hyponomy) is an asymmetric relation between two senses and also referred to as the **is-a** relation. The more specific sense (e.g. *bee*) is called a hyponym of the more general sense (e.g. *insect*), which is called hypernym. Jurafsky and Martin (2008) give a formal definition for Hyponomy in terms of entailment:

> "[...] a sense A is a hyponym of a sense B if everything that is A is also B and hence being an A entails being a B, or $\forall x\ A(x) \Rightarrow B(x)$." (Jurafsky and Martin, 2008)

Hypernomy is in most casses transitiv, thus if a *cow* is hyponym of *mammal* and *mammal* is a hyponym of *animal*, *cow* is also a hyponym of *animal*. an important phenomen for this thesis are two words, sharing a close hypernyms. In Figure 1, *bee* and *butterfly* share the hypernym *insect*, we refer to them as co-hyponyms.

2.2.3 Holonomy

Holonomy or Meronomy refers to the **part-whole** relation. In the illustration of Figure 1, the *wing* is a part of a *bird* and a *bird* is a part of a *flock*. We say that a *bird* is a meronym of *flock*, while *flock* is the holonym of *bird*. As opposed to Hypernyomy, this asymmetric relation is not automatically transitiv. While a *flock*[4] obviously consists of several birds, In this case *birds* is generally not replacable with *heads*.

2.3 Shortcut-Stacked-Encoder and Residual Encoder

We conducted most of our experiments with the Shortcut-Stacked-Encoder (Nie and Bansal, 2017) and the recently adapted version to the Residual Encoder for NLI. They achieve state-of-the-art results[5] for two large datasets[6] for NLI and follow the Siamese Architeture, originally introduced by Bromley et al. (1994), thus first encoding p and h using the same sentence encoder with shared weights into fixed length sentence representations and then predicting the entailment label from the combination of both representations using an additional Multi Layer Perceptron (MLP).

[4] This is an example for polysemy, as *flock* may refer to a group of birds, but also to a group of e.g. sheep. In this case we assume the sense of a group of birds.

[5] Considering models for SNLI without inter-sentence attention.

[6] These are explained in detail in §3.2

2.3.1 Sentence Encoding for Shortcut-Stacked-Encoder

The key novelty of this approach is the way sentence representations are created using a three-layer bidirectional Long-Short-Term Memory Network (biLSTM) with shortcut connections and row-wise max-pooling. An overview of this architecture is given in Figure 2. Due to the arbitrary length of words in textual input a widely used strategy to encode

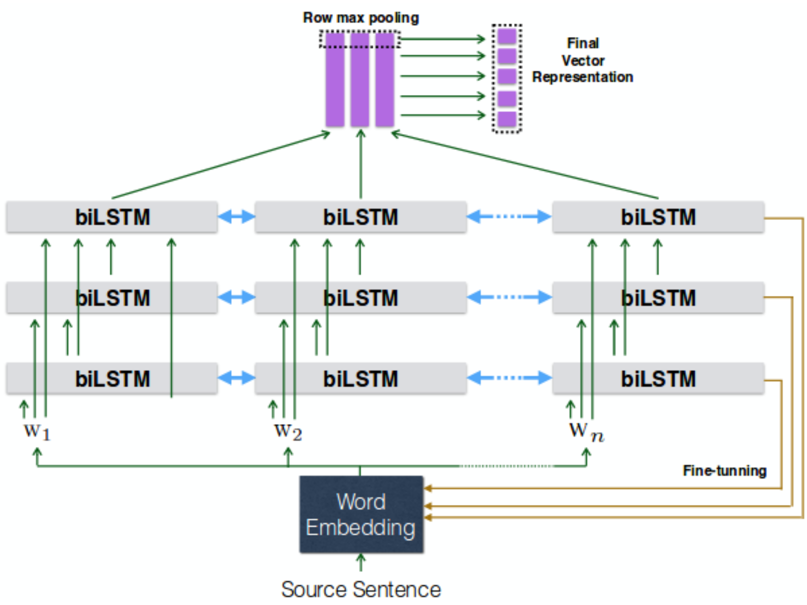

Figure 2: The architecture of the sentence-encoding component within the Shortcut-Stacked-Encoder, taken from Nie and Bansal (2017).

variable length inputs to fixed length vectors using LSTM (Hochreiter and Schmidhuber, 1997) or the bidirectional variant biLSTM (Graves and Schmidhuber, 2005). Essentially these components learn with the use of gates what information to keep and forget at a given point in time. By sequentially going through a sentence in one or two directions respectivly, are capable of exploiting word-order and take context into account.

The main difference of the Shortcut-Stacked Sentence-Encoder to typical approaches of a multi-layer biLSTM model is that the input to the biLSTM in a following layer is not only the output of the previous layer, but the output of *all* previous layers, together with the word embeddings. In the first step, the embedding layer maps each word ω_i of the source sentence $(\omega_1, \omega_2, ..., \omega_n)$ to a d-dimensional word vector $w_i \in \mathbb{R}^d$. According to Nie and Bansal (2017) we denote x_t^i to be the input of the ith biLSTM at timestep t. Naturally the input to the first layer are the word-embeddings itself, thus:

$$x_t^1 = w_t \tag{1}$$

In all biLSTM with $i > 1$ the input is the concatenation of all intermediate inputs of previous layers at the timestep t together with the initial word embeddings. Let [] denote the vector concatenation and h_t^i be the output of the ith biLSTM at timestep t, this leads to:

$$x_t^i = [w_t, h_t^{i-1}, h_t^{i-2}, ..., h_t^1] \tag{2}$$

Only the last biLSTM layer is used to generate the final sentence representation. Assuming m layers in total, d_m to be the hidden state dimension of the last layer, that is defined as $H_m = (h_1^m, h_2^m, ..., h_n^m)$ the final sentence representation v is obtained by applying row-max-ppoling over the last layer:

$$v = max(H^m) \tag{3}$$

With each $h_i^m \in \mathbb{R}^{2d_m}$ and $H^m \in \mathbb{R}^{2d_m \times n}$ the resulting sentence vector $v \in \mathbb{R}^{2d_m}$ essentiallly captures the highest value of each dimension over all timesteps[7].

[7] d_m is multiplicated by 2 since the biLSTM creates d_m features for going through the sentence forwards and backwards repsectively.

2.3.2 Classification

A two-layer MLP using ReLu as acticvation function and a final softmax-layer is used for the prediction. The input to the classifier m is the concatention of the sentence representations v_p and v_h for p and h respectively together with the element-wise distance and the elementwise product, denoted as $\otimes$ of both representations:

$$m = [v_p, v_h, |v_p - v_h|, v_p \otimes v_h] \tag{4}$$

Even thow a multi-layer MLP theoretically would be able to learn the latter two features, Mou et al. (2015) showed that this particular feature concatenation gives a performance gain for neural models for NLI.

2.3.3 Training

For all our reimplementations using pytorch[8] of the model we follow the parameters of the original paper of Nie and Bansal (2017). The model is trained using Adam (Kingma and Ba, 2014) parameter optimization, cross-entropy loss as objective function and minibatches of size 32. To avoid overfitting a dropout of 0.1 is applied on each layer of the MLP and the accuracy is evaluated regulary on a different dataset than the train data, the development set, as it is common practive in machine learning. The final performance is estimated by evaluating the best model based on the accuracy on the development set on unseen hold-out data, the test set. 300-dimensional GloVe 840B pretrained word-embeddings (Pennington et al., 2014) are used and finetuned during training. Three additional word-vectors are added, one for unknown words, as well as one to indicate the start and one to indicate the end of a sentence. The learning rate starts with 0.0002 and is reduced by half every second iteration. We conduct our experiments with different re-implementations of this model, partly due to using fewer parameters by reducing the dimensionality of the components, partly due to changes within the original paper.

2.3.4 Residual Encoder and Reimplementation Variants

In a second version of the paper, Nie and Bansal (2017) introduced the Residual Encoder, slightly adapting the way sentences are encoded. In order to create the input to the ith biLSTM layer x_t^i concatenation all previous outputs $(h_t^{i-1}, h_t^{i-2}, ..., h_t^1)$ together with w_t, naturally leads to a tremendous increase of parameters. By using residual connections, instead of concatenating all previous outputs, they are added up, thus equation (2) changes to

$$x_t^i = [w_t, h_t^{i-1} + h_t^{i-2} + ... + h_t^1] \tag{5}$$

and reduces the parameter size.

Implementation Variants

We use the following implementations of the model. The performance comparison between the models based on SNLI[9], is listed in Table 1 and do not differ tremendously from what Gong et al. (2017) estimated to be the human performance on the same task.

Model	SNLI train acc.	SNLI dev acc.	SNLI test acc.
Shortcut-Stacked Encoder†	87.4%	85.2%	84.8%
Shortcut-Stacked Encoder††	89.4%	86.0%	85.4%
Residual Encoder†	91.1%	85.9%	85.8%
Residual Encoder$^{\Diamond}$	91.0%	87.0%	86.0%
Human Performance (Gong et al., 2017)	-	-	87.7

Table 1: Accuracy in percent of different implementations of the model from Nie and Bansal (2017), achieved on the SNLI dataset compared with human performance.

- We refer to Shortcut-Stacked Encoder† as the first re-implementation. This uses 256×2, 512×2 and 1024×2 dimensions for the three layers of the sentence encoding biLSTM and 1600 dimensions in the classifier MLP.

[8] http://pytorch.org/

[9] SNLI is a huge dataset for NLI and will be explained in §3.2

- We refer to Residual Encoder† when we use our own re-implementation with residual connections. The sentence-encoding biLSTMs each have the dimensionality of 600×2 and the layers of the MLP of 800.
- We refer to Residual Encoder$^\Diamond$ when we use the final published version of Nie and Bansal (2017) with their provided code[10]. This model has the same parameter sizes as Residual Encoder†.
- We refer to the plain model name, when talking about the model structure in general.

[10] https://github.com/easonnie/ResEncoder

3 Related Work

Much work has been done to create strong models for NLI and we show some successful strategies in section §3.3 and relevant datasets in section 3.2. Before to the excessive usage of neural networks, many models heaavily relied on external resources, that have either been manually created in order to improve tools fpr NLP or arised for a different purpose, but can also be exploited. In §3.1 we show an overview of some external resources that might improve the performance of neural models on NLI. While most neural models rely only on distributed word-representations as external information and perform quite good, prior work (Bos and Markert, 2005; Tatu and Moldovan, 2005) heavily depended on resources. In §3.4 we show several approaches trying to combine the power of well structured, knowledge-rich resources with the generalization power coming from neural models with distributed word embeddings.

3.1 External Resources

A large variety of knowledge bases exist, containing for instance lexical relations or commonly known world knowledge, which can be helpfful for improving of NLI. Research has shown that both, manually created and automatically created resources, can successfully be applied in many tasks of NLP. Inn this section we only show WordNet and Wikipedia, containing different information that we consider to be useful for NLI and NLU, as well as two resources combining multiple resources and thus providing a huge amount of readily-available knowledge.

3.1.1 WordNet

WordNet (Miller, 1995) is a famous, manually created lexical resource for the English language consisting of three lexica for four different Part of Speech (POS), one for verbs, one for nouns and one for adjectives and adverbs respectively (Jurafsky and Martin, 2008).

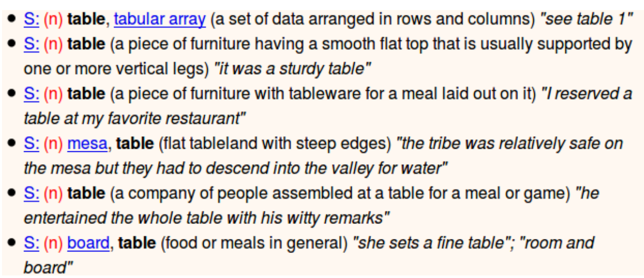
- S: (n) **table**, tabular array (a set of data arranged in rows and columns) *"see table 1"*
- S: (n) **table** (a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs) *"it was a sturdy table"*
- S: (n) **table** (a piece of furniture with tableware for a meal laid out on it) *"I reserved a table at my favorite restaurant"*
- S: (n) mesa, **table** (flat tableland with steep edges) *"the tribe was relatively safe on the mesa but they had to descend into the valley for water"*
- S: (n) **table** (a company of people assembled at a table for a meal or game) *"he entertained the whole table with his witty remarks"*
- S: (n) board, **table** (food or meals in general) *"she sets a fine table"; "room and board"*

Figure 3: Example of different synsets of the lemma "table" (only noun senses) within WordNet, taken from http://wordnetweb.princeton.edu.

Structure of WordNet

Mainly focusing on nouns[11] it differentiates between the more frequent class of *common nouns* like "table" and *instances* like "Berlin". All words are represented by their lemma and due to polysemy contain one or more senses, namely *synsets*. Synsets are the main building blocks within the WordNet ontology, containing a sense description and examples. Figure 3 displays 6 different senses for the lemma "table". It is noteworthy that the sense of table (as tabular array) greatly differs from the sense as furniture or tablelands while metaphorical senses strongly correlate with the sense of table as a furniture, encoding much more fine-grained sense-differences. While lemmata within the same synset refers to synonymy, other lexical relations like hypernomy, antonomy and holonomy (as described in Section 2.2, however more fine-grained[12]) are defined via labelled links between synsets. Thus, WordNet holds valuable knowldge for detecting lexical inferences in natural language.

Usage and Issues

When using WordNet in applications one has to identify the correct sense out of many possible synsets for a given lemma. This may be done using proper algorithms for Word Sense Disambiguation (WSD). Another simple and frequently used heuristic is to always choose the first snyset, which typically reflects the most common sense (McCarthy et al., 2004). As shown in Figure 3, word-senses are defined with different granularities, sometimes varying only with subtle differences that are not required by most applications. Subsequently this reduces the interpretability of path lengths of lexical relations between two synsets. For instance, identifying that "sunflower" is a hyponym of "plant" requires the traversal over five edges (*sunflower* → *flower* → *angiosperm* → *spermatophyte* → *vascular plant* → *plant*). At the same time, identifying that a "curch" is a "building" can be identified by only traversing over two edges (*church* → *place of worship* → *building*) and traversing similarily over five edges leads to the synset "whole, unit", covering both, living things and objects. This is a known issue (Resnik, 1995) and strategies exist to reduce the complexity of WordNet using sense clustering(Prakash et al., 2007).

[11] WordNet 3.0 contains 117,798 nouns, 11,529 verbs, 22,479 adjectives and 4,481 adverbs (Jurafsky and Martin, 2008).

[12] For example, WordNet differentiates between *hypernyms* for common nouns and *instance-hypernyms* for instances, or distinguished between *part-* , *ember-* and *substance-holonyms*.

3.1.2 Wikipedia

While WordNet contains manually annotated lexical relations and is easily automatically accessable, Wikipedia[13] is a huge multi-lingual, continuously growing encyclopedy, maintained by many volunteers. Also mostly focusing on nouns, due to the nature of containing ecyclopedic information, it contains a large variety of factual information about named entities, that many other lexical resources lack (Gurevych et al., 2016). While not being made for the purpose of being a lexical knowledge base, it may be seen as partially annotated due to artifacts like hyperlinks, thus can be interpreted similarily and even accessed using available tools in a programmatic manner (Zesch et al., 2008). Gurevych et al. (2016) describe the following information types that can be exploited to retrieve lexical information:

- The first paragraph of an article can be interpreted as the *sense definition*, since every article covers only one aspect by design.
- *Sense examples* can be retrieved from the context, surrounding a hyperlink that links to the entity of interest, showing how the term is used.
- Hyperlinks between articles can be considered as *sense relations*.
- Due to interlinked articles in different languages, the corresponding titles usually can serve as translation *equivalents*.

Wikipedia has successfully been used in many applications for NLP and even though we do not conduct experiments within this work using Wikipedia, it clearly contains rich factual and world knowledge that can be helpful for NLI systems.

3.1.3 Derived from multiple Knowledge Bases

Yet Another Great Ontology (YAGO) (Suchanek et al., 2007) combines the high coverage of Wikipedia with the clean taxonomy of WordNet, leading to a very knowledge rich resource. While YAGO mainly targets world-knowledge with Wikipedia being tremendously larger than WordNet and containing relations to express facts derived from it, UBY (Gurevych et al., 2012) aims for lexical semantic richness. In addition to Wikipedia and WordNet, seven other resources are combined together provoiding lexical semantic knowledge in German and English by so called *sense axis*, reaaltions betweem two senses of different lexicons. UBY provides an easy-to-use API, making its high coverage knowledge programatically accessable to NLP applications. Having these knowledge-rich resources avaivalbe, but for the most part decoupled from neural models, still lacking this knowledge, stresses the benefit of combining these two worlds.

3.2 Datasets for NLI

As neural models usually require a huge amount of data for their training, they were not successfully applicable to NLI tasks until the release of The Stanford Natural Language Inference Corpus (SNLI), reaching state-of-the-art results. Previous NLI tasks like FraCas (Cooper et al., 1996) or the PASCAL challenge (Dagan et al., 2006) only had very limited amount of training data, such that neural models could not be used. Some datasets like Sentences Involving Compositional Knowledge (SICK) (Marelli et al., 2014) or the Denotation Graph entailment set (Young et al., 2014) increased the amount of samples at the expense of using artificially created sentences and/or automatically labeling, reducing the textual quality and adding noise. Since the focus in this work is on neural models, only the relevant datasets for this purpose are introduced.

3.2.1 SNLI

With the release of SNLI (Bowman et al., 2015) researchers were able to apply neural models for the task of NLI using distributed word and phrase representations. The corpus consists of 570,152 human written sentence pairs and differentiates between the labels *entailment*, *neutral* and *contradiction*.

Event co-reference

A drawback of all previsously existing resources for NLI that is handled by Bowman et al. (2015), is the fact that even humans may assign different labels to a sentence pair, based on their subjective interpretation of a sentence, that all can be valid. This issue can be demonstrated using the following sentence pair:

Premise: *Young people are demonstrating in San Francisco.*
Hypothesis: *Young people are demonstrating in New York.*

[13] https://www.wikipedia.org/

One could clearly argue the sentence-pair should be labelled as *neutral*, since there could be people demonstrating in both towns. However it is also legimate to interpret these as contradicting sentences, if one considers both sentences to be describing the same event. While both sentences may be true when describing different potential scenarios, only one of them can be true if they refer to the same. In order to reduce noise coming from these inconsistent interpretations, the labeling scheme within SNLI must be fixed beforehand. Specifically they choose the labelling scheme to be based on event-coreference, the latter of the two explanations, as otherwise only very general statements would result in *contradiction*.

Generation

In order to create SNLI Bowman et al. (2015) used image captions from the Flickr30k corpus (Young et al., 2014) as premises and let human workers create according hypothesis for each label respectively using Amazon Mechanical Turk by asking them to write alternative captions that

- definetely also are a true description of the photo **(entailment)**
- might be a true description of the photo **(neutral)**
- definetely are a false description of the photo **(contradiction)**

The workers only saw the image caption, not the image itself, but were encouraged to use common world knowledge, enabling the creation of inferences that require additional information of the world, that is not available in word-embeddings[14]. While this process simplifies the task of assuming event-coreference, the sentences within SNLI are rather simple and short, due to the nature of image captions.

Looking into data

As we conduct most of the experiments of this work on SNLI it is important to get an understanding how sentences in this dataset look like. As previously mentioned the vast majority of sentences are rather simple and might even be phrases only rather than proper sentences due to omission of a verb. In addition to that, sentences might be written in proper English, but also might contain spelling or punctuation errors, be lowercased only or describe highly unrealistic scenarios. Table 2 shows selected sample sentence-pairs, taken from the SNLI dataset.

Premise	Hypothesis	Label
(1) The large brown dog jumps into a pond.	The dog is getting wet. The dog is a chocolate Labrador Retriever. A white cat is sunning itself on a windowsill.	*entailment* *neutral* *contradiction*
(2) A woman is handing out fruit.	A woman is passing out different types of fruits. A woman is handing out oranges. A fruit is handing out a woman.	*entailment* *neutral* *contradiction*
(3) A basketball game.	A sports game. A basketball game between rivals. A volleyball game.	*entailment* *neutral* *contradiction*

Table 2: Example sentence pairs, taken from SNLI, showing typical sentences within the dataset.

The first column displays the premise, the original image caption, in the second column three hypothesis are shown, created by the workers for each label respectively. Several characteristics of the dataset and types of required knowledge to solve the task can be seen here. The first examples **(1)** require the model to have some factual knowledge that a *Labrador Retriever* is some kind of *dog*, and *chocolate* is paraphrasing *brown*. Since *Labrador Retriever* is a possible substitute for *dog* but more specific, the sample is labelled as neutral. The according entailing hypothesis requires an even deeper understanding of the world, as the system needs to know that a *pond* is filled with water and anything that goes into water is *getting wet*. The contradicting sample shows two frequently occuring characteristics. Not only has *dog* been replaced by *cat*, but also the color and the activity changed. We found that in many contradicting hypothesis several contradicting words with respect to the premise exist, obviously making the task easier, as it would be sufficient

[14] For instance (taken from SNLI) *snow* is paraphrased as *frozen particles of water* and requires very deep factual knowledge to be understood correctly.

to only detect one indicator. Additionally it has been shown that the creation process of the hypothesis followed some unconscious heuristics of the worker(Gururangan et al., 2018). Specifically the replacement of *dog* to *cat* occurs often enough, that the presence of *cat* in the hypothesis alone is a strong indicator for contradiction already.

The sentences of the second example **(2)** are based on paraphrasing, representing the same meaning, (entailment), are more specific term in the hypothesis as in the previous example (neutral) and semantic role reversal (contradiction), which is somewhat interesting, as it requires to model to leverage word order, while a simple Bag of Words (BoW) approach would fail here.

In contrast to **(1)** the sentences in **(3)** only require very shallow knowledge. Here, the word *basketball* is substituted by it's hypernym[15] *sport*, thus still covering the original meaning by being more general. The next sentence gives some plausible additional information not given in the premise, hence neutral. In the last contradicting sentence the model has identify that *basketball* and *volleyball* are mutually exclusive, as generally one can only play on of the two sports at a time. As both terms are hyponyms of the term *sport* we refer to them as *co-hyponyms*. Amongst other relations like antonyms, co-hyopnyms are especially challenging for models using distributed word representations and will be a central point in section 5. While sentences are often longer than in this sample, the required knowledge specified in **(3)** is most present within SNLI.

3.2.2 MultiNLI

SNLI received some critizism within the research community (Chatzikyriakidis et al., 2017; Williams et al., 2017), mainly due to it's simplicity, cominng from the fact, that all premises are taken from a single genre only, namely image captions. Thus, SNLI is very limited to only visual scenes, neglecting many other aspects like temporal reasoning, modality or belief. Williams et al. (2017) introduced with MultiGenre Natural Language Inference Corpus (MultiNLI) a new dataset, overcoming these drawbacks.

Generation of MultiNLI

The authors followed the same generation procedure as has been done by Bowman et al. (2015), but instead of relying on image captions only, they took into considerations other genres from Open American National Corpus (OANC)[16] (Ide and Macleod, 2001; Ide and Suderman, 2004, 2006) as well as several freely available fiction work, resulting in 10 additional genres with 392,702 new sentence pairs for training and 20,000 for development and test respectively. A major motivation for the creation of MultiNLI was to put more emphasis on the role of NLI as evaluation benchmark of NLU that SNLI failed to provide due to its narrow coverage. Therefore only five of the new genres are present within the train data, while the remaining five genres only occur in the testing sets and serve for cross-domain transfer learning and domain adaption. The performance on this dataset is measured in two figures, *matched* examples are derived from the same source as training samples, while *mismatched* examples differ from those seen during the training. This motivation becomes also clear from the corresponding Shared Task (Nangia et al., 2017), allowing any kind of external resources (including the ones that were used to derive the premises) but only accepting sentence-encoding models[17] to evaluate sentence representations learning with respect to NLU. MultiNLI has been shown to be harder than SNLI (Williams et al., 2017), the best performing model of the RepEval 2017 Shared Task reaches 74.9% matched and mismatched accuracy (Chen et al., 2017c) using ensembles and 74.5% matched, 73.5% mismatched accuracy using a single model(Nie and Bansal, 2017).

Looking into data

Table 3 depicts a few samples of different genres[18]. One can see how different genres broaden the scope of language that is used to express inferences. A system needs to deal with temporal information and less visualizable terms like *appreciate* or *benefit*.

As the authors followed the guidelines used for SNLI and also assume event-coreference, both datasets are highly compatible, only differing in the range of genres and thus diversity of language. In fact, MultiNLI is even distributed in the same data format and a common practive is to include data from SNLI when training models for MultiNLI (Nie and Bansal, 2017; Balazs et al., 2017; Yang et al., 2017).

[15] At least in one sense. Hypernomy refers to the *is-a* relationship meaning *basketball*(hyponym) **is a** *sport*(hypernym). Word-relations are discussed in more detail in section 3.

[16] Genres from OANC: *Government, Slate, Telephone Speech, Travel Guides, 9/11 Report, Face-to-face Speech, Letters, Nonfiction Books, Magazine*

[17] These models encode each sentence individually and are explained in §3.3.

[18] Taken from https://repeval2017.github.io/shared/

Premise	Hypothesis	Label	Genre
The Old One always comforted Ca'daan, except today.	Ca'daan knew the Old One very well.	*neutral*	Fiction
Your gift is appreciated by each and every student who will benefit from your generosity.	Hundreds of students will benefit from your generosity.	*neutral*	Letters
At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	People formed a line at the end of Pennsylvania Avenue.	*contradiction*	9/11 Report

Table 3: Example sentence pairs from MultiNLI, taken from RepEval 2017 Shared Task, showing samples of different genres.

3.2.3 SciTail

SciTail (Khot et al., 2018) is yet another dataset for NLI, designed to address a different problem of previously existing datasets[19]. The targeted problem is of previous work is, that either the premise or the hypothesis was specifically for this task created, thus neglecting the kind of naturally occuring inference problems of any endtask like QA. It is comparably smaller, consisting of only 27,026 examples and only distinguishes between two labels, *entailment* and *neutral*. *Entailment* is defined as in SNLI and MultiNLI, saying that the premise supports the hypothesis. All cases where the hypothesis is not supported by the premise however are classified *neutral*.

Generation of SciTail

In order to retrieve premise and hypothesis from a resource rather than creating one sentence for the specific purpose of NLI, Khot et al. (2018) took a different approach to generate the corpus. The dataset originates from school-level multiple choice questions for science QA (Welbl et al., 2017). Those questions generally require sophisitcated reasoning capabilities in order to answer them correctly.

1. Given the short factual answer candidates, a new sentence was syntetsized using the question and correct answer, that serves as the hypothesis. For instance the question "When waves of two different frequen- cies interfere, *what phenomenon occurs?*" and the answer "beating" is transformed into "When waves of two different frequencies interfere, *beating occurs*" (Khot et al., 2018).

2. A large background corpus with relevant information from Clark et al. (2016) was used to generate candidate knowledge sentences for each question using Information Retrieval (IR) for the premise.

3. While hypothesis, derived from an icorrect answer, can be assumed to be not-supported by the premise, those derived from a correct answer are not necessarily supported by the sentence gained from the background corpus. Thus, samples where croudsourced annotated to ensure a correct labelling, only keekping those samples as *entailment*, that were labelled to have *Complete Support*[20].

Comparison with SNLI and MultiNLI

Due to it's design, SciTail is different in nature to the two previous datasets. Neither does it contain contradicting examples nor does assume event-coreference, as sentence-pairs in this dataset are more based on factual information. Table 4 shows somple sentences of the SciTail dataset. Clearly all of them contain factual information, whereas in the previous shown datasets, sentences tend to be more situational. The premise can be relevant for the entailment relation, yet must not be. In both cases the keywords *joint* or *range* appears in all premises, retrieved for both hypothesis respectively.

Due to its relatedness with Scientific QA, the authors claim, that a model reaching a good performance on this dataset for NLI will also score well on an according QA task, as similar NLU is needed.

[19] Ignoring small-scale datasets with less than 1,000 samples.

[20] Annotators could decide between *Complete support* (labelled as entailment), *Partial Support* (ignored) and *Unrelated* (labelled as neutral).

Premise	Hypothesis	Label
Bones come together to form joints, most of which are in constant motion.	Joints are the location where bones come together.	*entailment*
Bone, Joint, and Muscle Disorders Chapter 54 Charcot's Joints Charcot's joints (neuropathic joint disease) results from nerve damage that impairs a person's ability to perceive pain coming from a joint;	Joints are the location where bones come together.	*neutral*
The time to travel the horizontal distance (the range) is equal to twice the time to reach the peak (maximum height).	Range is the maximum horizontal distance traveled by a projectile.	*entailment*
First, finding the launch angle for maximum horizontal range in idealized projectile motion.	Range is the maximum horizontal distance traveled by a projectile.	*neutral*

Table 4: Example sentence pairs from SciTail Task, different premises retrieved for two hypothesis.

3.3 Neural Models for NLI

We follow the SNLI leaderboard[21] by differentiating between *sentence-encoding* and *inter-sentence-attention* based models, in the following we show an overview about relevant approaches of both areas. The Residual Encoder or Shortcut-Stacked Encoder introduced in section 2.3 belongs to the former class of models.

3.3.1 Sentence Encoding Models

Sentence-encoding models follow the Siamese Architecture (Bromley et al., 1994), meaning they encode both, sentences p and h, individually, with parameters being tied between both sentence encoders. The entailment classification is predicted by a following classifier like a MLP. Doing so, the models put more emphasis on a meaningful sentence representation with the motivation of being more generally applicable and less focused on the specifc characteristics of the task at hand (Bowman et al., 2016). Many different strategies are used to create meaningful sentence representations within this class of neural models. This is for instance done by exploiting syntactical information using neural Shift-Reduce-Parsers that creates a linear sequential structure from tree-structures sentence representations (Bowman et al., 2016) or by adding and external memory with read and write operations, capturing the temporal and hierarchical information within natural language (Munkhdalai and Yu, 2017).

Inner-Attention-based Models

Following the intuition that humans onyl remember certain parts of a sentence after reading it, Chen et al. (2017c) modeled this behavious using gated intra-sentence attention, by generating the sentence representation using pooling[22] strategies over the ouptouts of the encoding biLSTM, reweighted using the attention gates. The idea of using inner-attention mechanisms is also used by the best performing sentence encoders for SNLI, at the time of this writing reaching 86.3% accuracy (Shen et al., 2018; Im and Cho, 2017). Shen et al. (2018) create sentence representations using the combination of hard and soft self-attention[23]. While hard-attention forces the model to only focus on relevant elements of the input sequence, disregarding all others, it is not fully differentialbly and thus inefficient to train. Soft-attention methods on the other hand are fully differentiable and weights each element of the input sequence accoridng to their relevance. However by also giving positive, non-zero weights to irrelevant elements it diminishes the emphasis on truely important ones. By first applying hard-attention to retrieve a subset of context-aware elements that is afterwards processed using soft-attention, Shen et al. (2018) leverage the mentioned advantages both techniques. Inner attention is also used by Im and Cho (2017), however their model additionally uses directional masks, that prevent the network from considering following or predeeding words in the attiontion proces respectively and distance masks that reduce the attention weight if words are further away to each other. They show that their model outperforms others, especially with longer input sentences, as the result of considering word distance and positional information.

[21] https://nlp.stanford.edu/projects/snli/

[22] As done with max-pooling by Nie and Bansal (2017).

[23] A plain attention function calculates the alignment for an input sequence $x = [x_1, x_2, ..., x_n]$ given a query q. In the special case of self-attention, q arises from the input sequence x itself (Shen et al., 2018).

3.3.2 Inter-Sentence-Attention-based Models

Rocktäschel et al. (2015) showed that models perform significantly better when looking at both sentences simultaneously. This is motivated by the way humans would solve the task of NLI by first reading the premise and creating the understanding of the hypothesis based on the previously read sentence. Since this seems to be superior in the SNLI, many works followed this approach reaching state-of-the-art results. Also in this class of methods memory networks, accessable via attention were applied (Cheng et al., 2016).

Inter-Sentence-Attention-based models used within this work

Parikh et al. (2016) provide a simple network structure called Decomposable Attention, using the assumption that only parts of a sentence are needed for the entailment relationship. They do so by fragmenting the input sentences into subphrases and align fragments the fragments of both sentences with each other using attention. Even though they represent sentences in a BoW manner, they reach a remarkable performance. After comparing the aligned phrase pairs, the final sentence representation is retrieved by simple summation over the comparison-vectors, thus reducing without relying on any LSTM-based method and reducing computational complexity. Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017b) is another simple yet strong model, essentially consisting of three different steps. First, words are encoded using biLSTMs such that they represent the context as well as the word itself. Next, similarily to Parikh et al. (2016), they calculate the local inferance between elements in both sentences by reweighting the sentence representation based on the normalized attention weights and enhance this information using the feature concatenation of Mou et al. (2015), as done in the Shortcut-Stacked Encoder, however in this approach word order information is preserved by the network in contrast to Decomposable Attention. The final sentence representation is created using pooling[24] on the ouput of a biLSTM, composing the local inference information from the previous step, and is then fed into a classifier. While Decomposable Attention and ESIM achieve competitive results on SNLI we conduct experiments using both models Section 5, showing that these results are rather a matter of memorization than generalization.

Attempt to incorporate WordNet

Very recently Chen et al. (2017a) introduced with Knowledge-based Inference Model (KIM) a neural model incorporating information from WordNet, which at the time of this writing is the single best performing model on SNLI. In their approach, they map WordNet relations, as in Section 2.2, to a real number $0 \leq r \leq 1$, quantifying the relations between word within p and h based on their path length and represent each word-pair with this additional feature vector. This way they create new feature vectors yexternal resources only by a smal margin, ranging from 0.1 to 0.6 points in accuracy. Chen et al. (2017a) show that adding WordNet is helpful if less train data is available, however only show limited prove, that the model leverages from WordNet fused relations for the overall improvement in accuracy. In this paper we show that performance on SNLI is not sufficient evidence for the cpability of dealing with simple lexical inferences as inferred from WordNet, which suggests that further investigations should be conducted in this direction.

Benchmark

To this date the best single sentence-encoding models on SNLI reach 88.6% (Chen et al., 2017a) ensembles reach up to 89.3% (Tay et al., 2017; Peters et al., 2018; Ghaeini et al., 2018) giving an advantage of 2.3% or 3.0% respectively over the best sentence-encoding model (Im and Cho, 2017). Considering that the estimated human performance on SNLI only is estimated to be 87.7% (Gong et al., 2017) indicates, that research started to slightly overfit on the dataset already.

3.4 Integration of external Resources into Neural Networks

There have been several approaches to integrate knowledge of different kind (as described in Section §3.2) into neural networks. Hu et al. (2016) infer external knowledge represented in logical form using a student-teacher setup, with the teacher being a neural network constrained by the rules acquired from an external resource, the student being a neural networkconsidering the true labels and the constrained predictions of the teacher. By simultaneous training both networks influenced by each others predictions, the integrated within the networks, weighted by their learned relevance. Most attempts however target improving word representations.

3.4.1 Improving word-embeddings

A very intuitive way to integrate external resources is to enrich word-embeddings with additional information. As most neural models depend on those anyway, any imrovement of word-representations can be adapted with very limited effort.

[24] As opposed to summation in Decomposable Attention. Chen et al. (2017b) evaluate in their experiments, that pooling leads to superior results than summation, due to being less sensitive to the sentence length.

Joint learning of distributional embeddings with external information

Xu et al. (2014) differentiate between *categorical* (attributes of words like “location”) and *relational* (relations between words like “child-of”) knowledge and train the embeddingss from scratch using three objective functions simultaneousl. They use skip-gram to encode distributional properties, minimize the distance between words sharing the same category and represent a relation as a vect or r, such that for a word w_1, connected to another word w_2 via relation r the equation $w_1 + r \simeq w_2$ holds (evaluated on analogical reasoning, word similarity, topic prediction). Liu et al. (2015) construct enriched embeddings using a constrained optimization problem. Specifically, they create constraints by ranking word similarities such that for instance synonyms should be more similar than antonyms or hyponyms should be more similar to close hypernyms than to distant hypernyms and include these into the training process with skip-gram.

Post-processing existing representations

Faruqui et al. (2015) propose a method called *Retrofitting* (evaluated on lexical semantic evaluation tasks), a post-processing method than can be applied on any pre-trained word-representations. They reduce the eucledian distance between words that are connected with a lexical semantic relation within a resource, while also keeping the representations close to the original neighbouring word-representations. Attract-Repel (Mrkšić et al., 2017) is another retrofitting method, essentially pulling synonyms closer to each other while pushing antonyms apart in vector space while keeping the original distributional information. Similarily Vulić and Mrkšić (2017) build on attract-repel adding hypernym relations for the context of lexical entailment by using an asymmetric distance measure between hypernym-hyponym pairs.

Effectiveness of iproved embeddings

The demand of integrating lexical resources such as WordNet has mainly been targeted by enrichening word-represnetations, with previously mentioned approaches being just a small selection. The improvement of previously explained approaches however is either demosntrated by visualizing word-relation vectors, that may not even be expoited by end-to-end neural networks (Levy et al., 2015), or based on evaluations on very low-level tasks like Word-Similarity, Syntactic Relations or Analogical Reasoning, or solely provide intrinsic evaluation. Neural networks for higher level tasks like NLI however reach state-of-the-art performances still relying only on standard pre-trained distributional word-representations like GloVe, even though alternatives exist, and premimitary experiments[25] of using enriched embeddings for SNLI have shown no success. We evaluate the possibility of adding enriched embeddings following the idea of Rücklé et al. (2018), that different embeddings encode complementary information and thus concatenating them. However we focus our experiments on the integration of knowledge on a higher level, the sentence representation, due to limited reported success on end tasks using richer embeddings, though many of those embeddings exist.

[25] These experiments have been conducted by Vered Shwartz in prior work and are not part of this work.

4 Understanding Shortcut-Stacked-Encoder

In this section we give analyse the sentence representations of Shortcut-Stacked Encoder† by visualizing how they encode (Section §4.2) and leverage (Section §4.3) information from natural language text, coming from SNLI. Additionally we show experiments, underlining the presented insights.

4.1 Motivation

The major downside of neural networks is the lack of interpretability (Goldberg, 2017), thus their capabilities on a lower level can only be estimated by finding meaningful evidence for their failures or sucesses on the task at hand. While analysing errors may lead to conclusions *what* does not work, *why* it does not work is in many cases left to intuition. Other machine-leanring classes like probabilistic or symbolic techniques do not suffer from this problem, leading to an increasing interest in visualization techniques for neural networks. Most visualitations of sentence-representations to date focus on attention-based approaches showing how words are aligned to each other such as by Shen et al. (2018) or Im and Cho (2017). To the best of our knowledge, no insights have been gained to understand the final sentence representation in vector space. In this section we demonstrate how this representation, arising from max-pooling, can be interpreted, using the Shortcut-Stacked Encoder† as the model to analyze. Intuitively, understanding how the Shortcut-Stacked Encoder† encodes information can be helpful for the task at hand of improving it using external resources.

While we did not manage to leverage the insights gained in this chapter to increase the performance, it might be helpful for future work. NN nicht gut interpretierbar

4.2 Insights on the sentence representation

In this section we show how we analyse the informaation that is present within the sentence represions, what kind of information is encoded and demonstrate, that the sentence representation can manually be adjusted in a meaningful way.

4.2.1 Approach

We use Shortcut-Stacked Encoder†, trained on SNLI, for our analyses. This model creates for input each sentence x, cosnisting of words, represented as x_i, a sentence representation $r \in \mathbb{R}^{2048}$ with r_j being the jth dimension of r. Arising from x, r captures the relevant information for the task at hand and is used in many neural networks without a deeper understanding what each r_j actually encodes. We shed light into the dimensionewise meaning of the sentence represnetation by identifying which word is responsible for the actual value of r_j.

Method

For simplicity, We explain our applied method and the reason why we use Shortcut-Stacked Encoder using a more general neural architecture of LSTMs, a simple uni-directional RNN. Figure 4 (left) shows the recursive workflow of such a RNN, following the notations of Goldberg (2017). Maintaining an internal state $s \in \mathbb{R}^m$, for m-dimensional representations, the

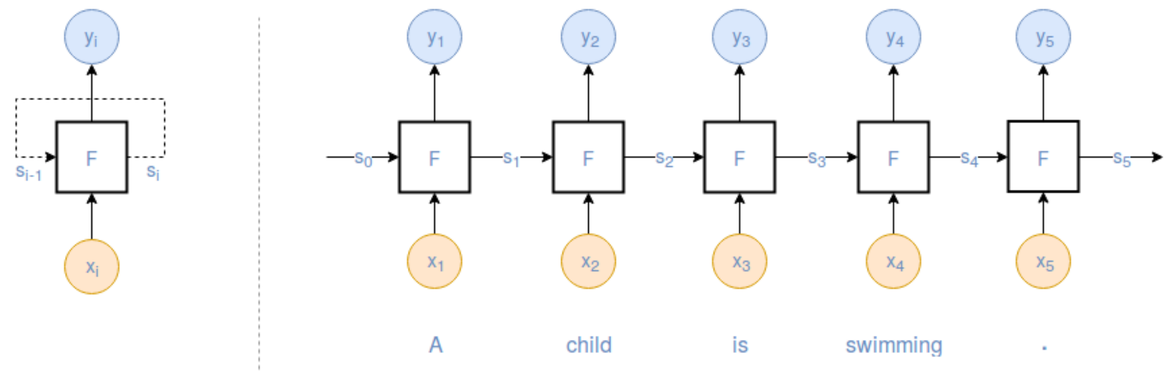

Figure 4: General architecture of a RNN (left). Example sentence in an unrolled RNN (right).

network recursively iterates over the input sequence x, aggregating in each timestep the previous state $s_{i-1} \in \mathbb{R}^m$ with the current input x_i using the function F. This state is then used for the next iteration and output via a mapping function as $y_i \in \mathbb{R}^m$. Multiple implementation variants exist of F and what is shared across iterations. LSTMs for instance use several neural gates to learn what information should be used, output or forgotten. This procedure ca be seen with an example sentence yb unrolling the network in Figure 4 (right). In typial setups a neural network may either choose to use s_t or y_t for a sequence length of t as the final sentence representation (Goldberg, 2017), since the network iterated over the full input sequence and contains the relevent information, if optimized for it. Even though the architecture of different versions of RNN may be well understood and has a logical meaning, the actual procedure of deriving concrete representations within a trained model is hard to understand. We leverage the fact that the Shortcurt Stacked Encoder uses max-pooling over all y_i to gather the sentence representation rather than using y_t or s_t by identifying what y_t has the highest value within a given dimension and mapping this dimension to the word x_t of the input sentence. As an example consider the sentence in Figure 4 (right). For each timestep t a new vector y_t is produced. As done by Nie and Bansal (2017) we concatenate all y_t to a matrix $\mathbb{R}^{m \times t}$, with m being the representation size and each vector y_t being the tth row within M. Assuming a dimensionality of $m = 3$, an examplatory matrix M for the given sentence "A child is swimming ." is displayed in Figure 5. Additiobnally to creating the sentence representation r by applying wor-wise

$$M = \begin{bmatrix} 1 & 4 & \mathbf{7} & 2 & 0 \\ 2 & \mathbf{9} & 4 & 1 & 1 \\ 0 & 3 & 2 & \mathbf{8} & 2 \end{bmatrix} \xrightarrow{\text{argmax}} r = \begin{bmatrix} 7 \\ 9 \\ 8 \end{bmatrix} \; a = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} \xrightarrow{\text{map}} \begin{bmatrix} \text{is} \\ \text{child} \\ \text{swimming} \end{bmatrix}$$

Figure 5: Visualized example of extracting interpretable information of the max-pooled seentence representations with a dimensionality of 3.

max-pooling on M, we collect the vector a, containing the column indizes, that are responsible for the values within r. These can directly be mapped to the word of the source sentence and thus be interpreted by humans. It should be notted that due to the nature of the multi layer biLSTM each y_t does not only contain the word at x_t but its context. While this somehow may lead to less accurate mapping, we found that the chosen method is sufficient to gain some meaningful insights on sentence encoding.

Analysed data

To reduce noise and aming for sentences that Shortcut-Stacked Encoder† seems to have a proper understanging about, we sample 1000 sentence representations from the SNLI train data in the following strategy. We group all sentence pairs (p, h) sharing the same premise and only keep groups if all samples belonging to the same group are classified correctly. Thus, we reduce the amount of sentences that are definetly misunderstood by the model, that would be harder to interpret. For now we are not interested in the actual relation between p and h and therefore create a pool of the remaining sentences, by treating p and h equally and splitting their connections apart. After removing duplicate sentences, the most frequent sentence length for the remaining representations is 8. To reduce noise that may arise from different sentence lengths, we only consider sentences of a length of 8 and randomly sample 1000 sentence representations. All experiments in this chapter are based on the same instances, unless otherwise stated.

In addition to the representation values each sample contains the following information:

- **Token:** The tokens that triggered the maximum value for the representation.
- **Token position:** Positional information about the responsible tokens within the sentence.
- **Lemma:** The lemmata of the responsible tokens.
- **POS:** The POS tags of the responsible tokens.
- **Dependency Parse:** The tags of the responsible tokens within the dependency parse tree.

Lemmatizing, POS-Tagging and dependency parsing were conducted using spaCy[26].

4.2.2 Detection of relevant dimensions

As commonly done when analysing data we start by showing a rough look into the sentence representations at hand. Typically, the Standard Deviation (SD) within a dimension correlates with the with the relevance for decision making.

[26] https://spacy.io/

Naturally, a dimension that does not change its value and thus being close to a constant is not informative, while a value with a high SD can be considered informative (Bishop, 2007). We calculate SD over all dimensions, depicted as a histogram in Figure 6. We plot the standard deviations in a discrete space using a bin size 0.05. For each if the 2048

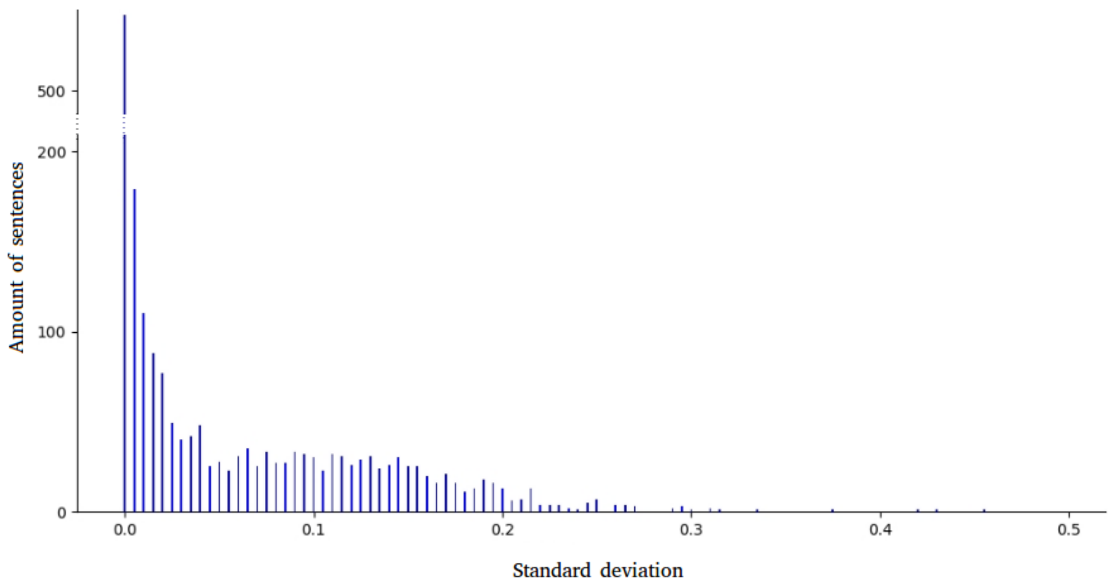

Figure 6: The standard deviation within a dimension of sentence representations (x-axis) by the amount of dimensions with the given standard deviation.

dimensions we calculate its SD to assign them to the correct bin. The amount of dimensions with the given SD is shown on the y-axis, note that the upper part of the plot is truncated for the sake of compactness. As can be seen, only a very tiny fraction of the dimension shows a large variation, the vast majority contains more or less the same value, regardless of the sentence. This obviously does not mean, they contain no information at all, as they may only be used to encode information that is rarly present within the data, however itserves as a reliable source, what dimensions are relevant to the model.

A naive approach to identify dimensional encoding

An intuitive approach to identify, what is encoded within the sentence representation, is to find common similarities between the words across all sentences, that are responsible for the according dimmension. Especially the task of NLI we assume *semantic*, *syntactic* or *positional* information to be required. Those can all be inferred using the features we extraced in Section §4.2.1. Similarities between words heavily depend on the context they appear in (Dagan, 2000). For instance one could consider a car and an identical recunstruction in original size of the same car as similar, whereas a horse is very distinct. Adding additional information that one needs to reach a destination in short time, he or she is more likely to consider the horse similar to the car, desicing between these two option. This essentially comes to a major problem when investigating semantic encoding without prior knowledge of what attributes may be considered relevant. We therefore investigate the sentence representation using excessive manual analyses in a top down manner, by first searching for patterns across all dimensions. In Section §4.2.3 we will look into some dimensions in detail.

A tool for sentence representation visualization

In order to evaluate many patterns with minimal time effort, we create a visualitazion tool, capable of dynamically generating any labelling scheme for responsible words based on the features descried previously. A sample visualitation is shown in Figure 7. This grid-plot visualizes for each row the responsible words for one dimension, listed on the right side as (<rank in terms of SD[27]>, <dimension index>), colored based on the attributes of interest. In this particular case words are colored by their position within the sentence. Each column refers to the same sentence along different dimensions. As a trade-off between explanatory power and clarity we always plot 300 sentences on 300 dimensions, which are eigther oredered by SD or already pre-sorted by the frequency[28] of a label of interest. In this particular case,

[27] All dimensions are ranked by their SD, giving an intuition of the expressiveness of the dimension.

[28] Even though we only use 300 sentences and dimensions for plotting, calculatiions are based on all the selected data.

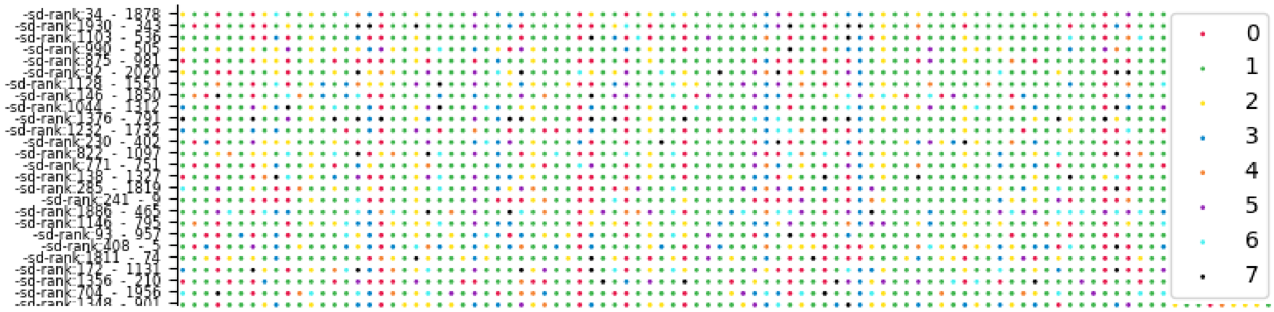

Figure 7: An extraction of a grid-plot, showing dimensions with the position within the sentence of the word, responsible for the dimensional value.

dimensions are ordered by their frequency of words on the üposition 1, meaning the upmost dimension received its values from the second word (index 1) more than any other dimension. Looking for patterns across many sentences, we focus on horizontal lines with the same coloring. Vertical lines indicate different differences across sentences w.r.t. the attribute of interest.

Filtern + Select by SD, Search

create tool label data and sort by label freqency + SD + dimension positions

responsible word i nitcht genug, value is immer wichtig

4.2.3 Dimension-wise Analysis

4.2.3.1 Positional information

4.2.3.2 Semantic information

4.2.3.3 Syntactic information

4.2.3.4 Evaluation of the impact of female and male dimensions

4.2.4 Conclusion

weg??

4.3 Insights on the sentence alignment

4.3.1 Approach

4.3.2 Entailment analysis

4.3.3 Neutral and contradiction analysis

4.3.4 Experiments

4.3.5 Conclusion

- eher experimental, need different models w maxpooling, mehr daten, mehr experiments, ...

4.4 Errors of the base model

5 Additional SNLI test-set

5.1 Motivation

5.2 Dataset

5.2.1 Creation

Upward/Downwar monoton (MacCartney and Manning, 2007) (?) https://www.illc.uva.nl/Research/Publications/Reports/PP-2008-05.text.pdf

5.2.2 Validation

5.2.3 Final dataset

5.3 Other models

5.3.1 ESIM

5.3.2 Decomposable Attention

5.4 Evaluation

5.5 Analysis

6 Approaches to incorporate WordNet information

6.1 Extraction of WordNet data

6.2 Integrating information into word-embeddings

6.2.1 Motivation

(Rubinstein et al., 2015) show something that distributional embeddings not always good (reread)

6.2.2 Concatenating pre-trained word-embeddings

6.2.3 Concatenation categorical information

6.2.4 Analysis

6.3 Multitask Learning

(Levy et al., 2015) show that embeddings not necessarily (reread)

6.3.1 Motivation

6.3.2 Architecture

6.3.3 Approaches

6.3.3.1 Different sizes of multitask MLP

6.3.3.2 Introducing Dropout

6.3.3.3 Introducing an additional shared layer

6.3.3.4 Fixing multitasking network during training

6.3.3.5 Focusing on original words within sentence representation

6.3.4 Analysis

6.3.5 Evaluation

7 Conclusion

References

Jorge A Balazs, Edison Marrese-Taylor, Pablo Loyola, and Yutaka Matsuo. 2017. Refining raw sentence representations for textual entailment recognition via attention. *arXiv preprint arXiv:1707.03103*.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Christopher M. Bishop. 2007. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 628–635. Association for Computational Linguistics.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744.

Asli Celikyilmaz, Dilek Hakkani-Tur, Panupong Pasupat, and Ruhi Sarikaya. 2010. Enriching word embeddings using knowledge graph for semantic tagging in conversational dialog systems. *genre*.

Stergios Chatzikyriakidis, Robin Cooper, Simon Dobnik, and Staffan Larsson. 2017. An overview of natural language inference data collection: The way forward? In *Proceedings of the Computing Natural Language Inference Workshop*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, and Diana Inkpen. 2017a. Natural language inference with external knowledge. *arXiv preprint arXiv:1711.04289*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017c. Recurrent neural network-based sentence encoder with gated attention for natural language inference. *arXiv preprint arXiv:1708.01353*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter D Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*, pages 2580–2586.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Ido Dagan. 2000. Contextual word similarity. *Handbook of Natural Language Processing*, pages 459–475.

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.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615.

Reza Ghaeini, Sadid A Hasan, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Z Fern, and Oladimeji Farri. 2018. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. *arXiv preprint arXiv:1802.05577*.

Yoav Goldberg. 2017. *Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies, Band 37)*. Morgan & Claypool Publishers.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M Meyer, and Christian Wirth. 2012. Uby: A large-scale unified lexical-semantic resource based on lmf. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 580–590. Association for Computational Linguistics.

Iryna Gurevych, Judith Eckle-Kohler, and Michael Matuschek. 2016. Linked lexical knowledge bases: Foundations and applications. *Synthesis Lectures on Human Language Technologies*, 9(3):1–146.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. 2016. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679.

Nancy Ide and Catherine Macleod. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics*, volume 3, pages 1–7. Lancaster University Centre for Computer Corpus Research on Language Lancaster, UK.

Nancy Ide and Keith Suderman. 2004. The american national corpus first release. In *LREC*.

Nancy Ide and Keith Suderman. 2006. Integrating linguistic resources: The american national corpus model. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*. Citeseer.

Jinbae Im and Sungzoon Cho. 2017. Distance-based self-attention network for natural language inference. *arXiv preprint arXiv:1712.02047*.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing, 2nd Edition*. Prentice Hall.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

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

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1501–1511.

Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the conference on empirical methods in natural language processing*, pages 802–811. Association for Computational Linguistics.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Association for Computational Linguistics.

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. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Using automatically acquired predominant senses for word sense disambiguation. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422*.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association of Computational Linguistics*, 5(1):309–324.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural semantic encoders. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 1, page 397. NIH Public Access.

M Lynne Murphy. 2003. *Semantic relations and the lexicon: Antonymy, synonymy and other paradigms*. Cambridge University Press.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172*.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. *arXiv preprint arXiv:1708.02312*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Rion Snow Sushant Prakash, Daniel Jurafsky, and Andrew Y Ng. 2007. Learning to merge word senses. *EMNLP-CoNLL 2007*, page 1005.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence-Volume 1*, pages 448–453. Morgan Kaufmann Publishers Inc.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 726–730.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated p-mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv:1803.01400*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. *arXiv preprint arXiv:1801.10296*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Marta Tatu and Dan Moldovan. 2005. A semantic approach to recognizing textual entailment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 371–378. Association for Computational Linguistics.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*.

Ivan Vulić and Nikola Mrkšić. 2017. Specialising word vectors for lexical entailment. *arXiv preprint arXiv:1710.06371*.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. *arXiv preprint arXiv:1706.00377*.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1219–1228. ACM.

Han Yang, Marta R Costa-jussà, and José AR Fonollosa. 2017. Character-level intra attention network for natural language inference. *arXiv preprint arXiv:1707.07469*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *LREC*, volume 8, pages 1646–1652.

List of Figures

List of Tables