Understanding and Improving Neural Models for Natural Language Interference using External Resources

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

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Physik
Institut f"ur Angewandte Festkernphysik
Speerspitze der Elite

Understanding and Improving Neural Models for Natural Language Interference using External Resources
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:

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
HIT	Human Intelligence Task
IE	Information Extraction
IR	Information Retrieval
KIM	Knowledge-based Inference Model
LSTM	Long-Short-Term-Memory
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
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
ReLU	Rectified Linear Units
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., 2013a) 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 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 NLI, thetask that is used within this work. We also explain the architeecture of the model, that is used within most experiments and define lexical relations, that play an important role within this thesis.

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, in the remainder of this thesis denoted as h, is said to be entailed by the premise, denoted as p, if a human reader would conclude, that h is true, given the fact that the p is true. This definition differes from strict logical inference in the following way: While in NLI *a high plausability* for the p to imply the h, based on the human judgement, is sufficient, the strict logical inference strives to achieve *certainity* (Dagan et al., 2009). NLI essentially breaks down to an alignment problem (MacCartney et al., 2008), shown in the following example. 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. Following Bowman et al. (2015), the sentence relation can be classified using one out of three labels, *entailment*, *neutral*, *contradiction*. Examples, taken from the SNLI Leaderboard[1], for each label are shown in Table 1. If a human can infer that h is very likely to be true, given the fact that p is true, the gold label is entailment. In the first example, the hypothesis describes men “playing sport”, which amongs other includes playing the “soccer game”, and thus definetly still holds. In the second sentence pair, both sentences describe two smiling man, however the hypothesis adds information, that they are smiling “at the cats”. While this new information may be true, it only is one of many potential scenarios and unkown, given the premise, thus the sample is labelled as neutral. Ih h cannot be true if p is true, the label is contradiction. In the last example, obviously the man cannot “inspect” anything and “sleep” at the same time, thus the labelling as contradiction.

Sentence-pair	Gold Label
A soccer game with multiple males playing. Some men are playing sport.	entailment
An older and younger man smiling. Two men are smiling and laughing at the cats, playing on the floor.	neutral
A man inspects the uniform of a figure in some East Asian country. The man is sleeping.	contradiction

Table 1: Example sentence-pairs for each possible label, taken from SNLI Leaderboard
.

2.1.1 Relatedness to other NLP tasks

While NLI clearly is central to computational reasoning capabilities, as it detects the inference relationship between two texts, it is also very fundamental and applicable to a large variety of NLP tasks, as the ability to recognize textual entailment is a fundamental and necessary problem towards real NLU (MacCartney and Manning, 2007; Bos and Markert, 2005) in general. Many NLP applications such as Question Answering (QA), Summarization or 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 (Dagan et al., 2009). All three tasks require the model to infer, that the target meaning of interest can be inferred from corresponding other variants, consisting of a different textual expression. For QA this is related to the identification of a correct answer. For summarization, on the one hand, the complete summary needs to be implied by the original text, on the other hand redundant sentences expressing the same meaning, thus one impying the other, should be ommited. Similarly IE, especially if using multiple documents, needs to infer, whether two variants of text contain the same information, thus sentences entailing each other, or not. 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

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

capabilities. Thus, one of the main purposes of NLI, being a very basic problem towards NLU, is serving as a benchmark to directly improve NLU, with any enhancement potentially helping a large variety of higher level tasks within NLP (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[2], whereas *Lexical Semantic Relations* are a special form of lexical relations, consisting of relations, that refer to the meaning of the word (Murphy, 2003), which have shown to be helpful for detecting lexical inferences (Dagan et al., 2009). We define those relations in this subsection based on the definition of Jurafsky and Martin (2008). One key characteristic of natural language is ambiguity, which also is present in lexical semantics as words may have several meanings or *senses*[3]. 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[4] in Figure 1.

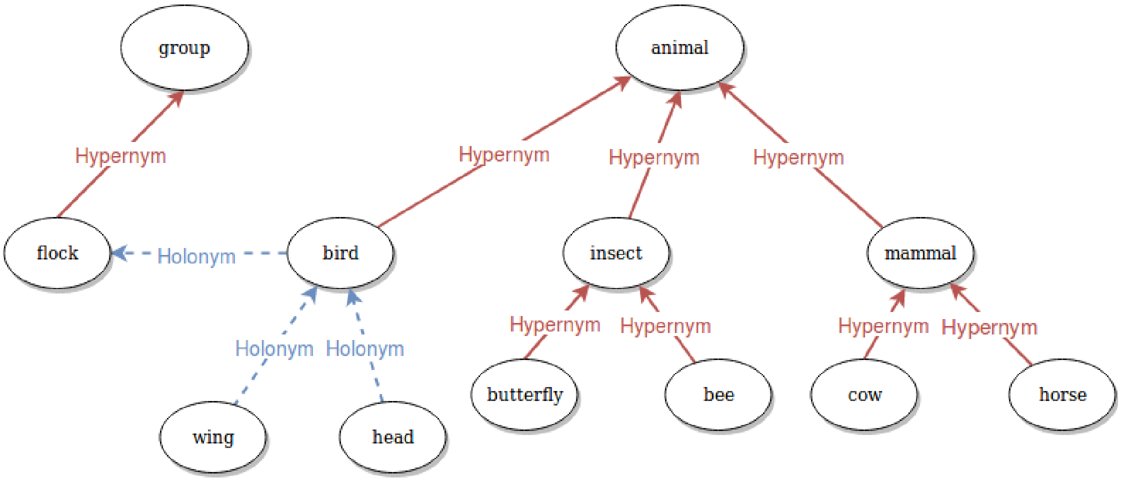


Figure 1: A sample ontology of animals to illustrate the lexical relations *Hypernomy* and *Holonymy*.

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 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, antonomy is a symmetric relationship between senses. These senses however have 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 the *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:

[2] While there is no single definition for *word*, we use the term equivalent to a single token identified by typical tokenizers, thus it is defined by its surface form.

[3] The phenomen of words having multiple senses is called *homonymy*, if both senses have no meaningful relation but still 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)

[4] 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*.

> "[...] 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)

In most cases, hypernomy is transitiv, thus if a "cow" is a hyponym of "mammal" and "mammal" is a hyponym of "animal", "cow" is also a hyponym of "animal". An important relation for this thesis holds between two words, sharing a close hypernym. In Figure 1 for instance, "bee" and "butterfly" share the close 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 generally transitiv. While a "flock"[5] obviously consists of several birds, In this case "birds" is generally not replacable with "heads", even though "head" is a meronym of "bird".

2.2.4 Lexical semantic realtions for NLI

the introduced lexical semantic relations are far from complete. One may define many other relations that hold between two words, like for example *president-of* can define the relationship between "Donald Trump" and "USA". Yet, the presented relations are well captured in various lexical knowledge bases, that will be explained in Section §3.1, and even though they only capture a small amount of the requirements for NLI, identifying those relation amongst words of p and h is a crucial for the task (Shwartz et al., 2015). Even though they exclude phenomenas like causality, at the very basic, a model for NLI should identify that synonyms or hypernyms of a word cover the same meaning an thus can be inferred. While not always, similar indicators are given by meronyms, here however this may differ, depending on the sense itself or its context (Shwartz et al., 2015). Meronyms for locations are usually covered by their holonym. For instance "John is in *Paris*" implies "John is in *France*", with "Paris" being a meronym, or *part-of* "France". However for the example in figure 1, the opposite holds: "A lion eats a *flock*" implies that the lion eats a "bird". As opposed to the locations, in this case, the holonym "flock" covers the meronym, not vice-versa. In the remainder of this thesis, we refer to the presented lexical semantic relations, applied on the entailment problem, when referring to *lexical inference*.

2.3 Shortcut-Stacked-Encoder and Residual Encoder

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

2.3.1 Sentence Encoding for Shortcut-Stacked-Encoder

The key novelty of this approach for NLI 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 amount of words in textual input, a widely used strategy to encode variable length inputs to fixed length vectors is the usage of LSTM (Hochreiter and Schmidhuber, 1997) or the bi-directional 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, meaning at a given word in sequential order within a sentence, when applied to text. By sequentially going through a sentence in one or two directions respectivly, these eural components are capable of exploiting word-order and take context of each word into account, when creating the compact sentence representation.

The main difference of the Shortcut-Stacked Encoder to typical architectures, using a multi-layer biLSTM, is, that the input to the biLSTM in a following layer is not only the output of the previous layer (as commonly done), but the output of *all* previous layers, together with the word embeddings. This is visualized within Figure 2 and referred to as "Shortcut-connections" (Nie and Bansal, 2017). Let t denote the word position at the current time step within the input sentence, consisting of a total of $n$words. In the first step, the embedding layer maps each textual word ω_t with $t \in \mathbb{N}$ and $0 < t \; leqn$ of the source sentence $(\omega_1, \omega_2, ..., \omega_{n-1}\omega_n)$ to a d-dimensional word vector $w_t \in \mathbb{R}^d$. According to Nie and Bansal

[5] 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 and ignore other senses.

[6] These are explained in deeper detail in §3.2

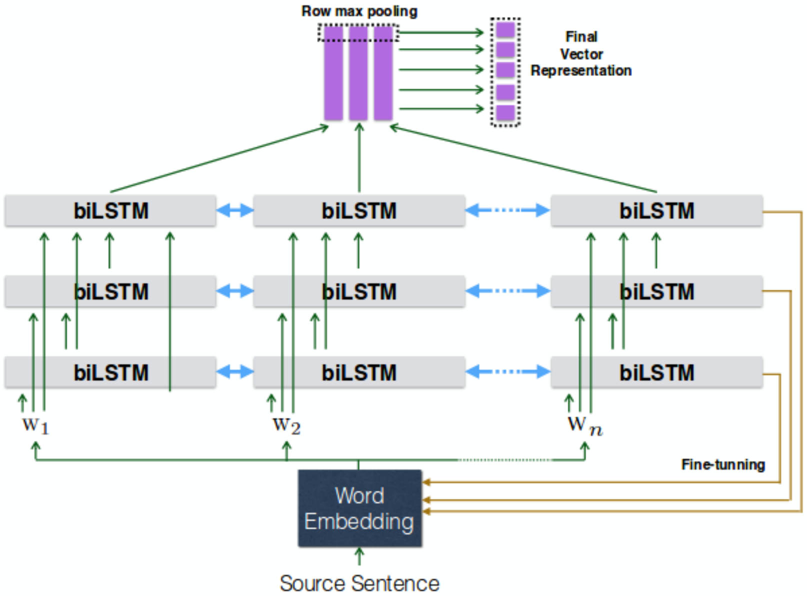


Figure 2: The architecture of the sentence-encoding component within the Shortcut-Stacked-Encoder, taken from Nie and Bansal (2017).

(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 w_t. Let $[]$ denote the vector concatenation and h_t^i be the output of the ith biLSTM at timestep t. The input x_t^i formally looks as follows:

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

Only the last biLSTM layer is used to generate the final sentence representation. Let m be the amount of layers in total and d_m be the hidden state dimensionality of the last layer, that is defined as $H_m = (h_1^m, h_2^m, ..., h_n^m)$, thus a matrix, consisting of all aligned outputs h^t of the last layer. The final sentence representation v is obtained by applying row-max-pooling over the last layer's output vectors:

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

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]. We strongly leverage from the max-pooled sentence representation in Section §4 and discuss how this method can be exploited in the corresponding section in detail.

2.3.2 Classification

A two-layer MLP, using Rectified Linear Units (ReLU) as activation function, and a final softmax-layer are used for the prediction. The input m to the classifier is the concatention of the sentence representations v_p and v_h for p and h respectively, together with the element-wise difference and the element-wise product, denoted as $\otimes$, of both representations:

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

Even though a 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.

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

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 on a regular basis on a different dataset than the train data, the development set, as it is common practice 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 fine-tuned 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 reduce training time 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 same paper, Nie and Bansal (2017) introduced the Residual-Stacked Encoder, slightly adapting the way sentences are encoded. Creating the input to the ith biLSTM layer x_t^i by concatenating 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 with an increasing amount of layers. By using residual connections, instead of concatenating all previous outputs, previous outputs are summed up instead of being concatenated, 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 2 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 2: 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 with reduced parameter size w.r.t. the original proposed model. 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.
- We refer to Shortcut-Stacked Encoder†† as the the same re-implementation using the full parameters, as reported originally. This uses 512 × 2, 1024 × 2 and 2048 × 2 dimensions for the three layers of the sentence encoding biLSTM and 1600 dimensions in the classifier MLP.
- 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†.

[8] http://pytorch.org/

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

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

- We refer to the plain model name, when talking about the model structure in general.

3 Related Work

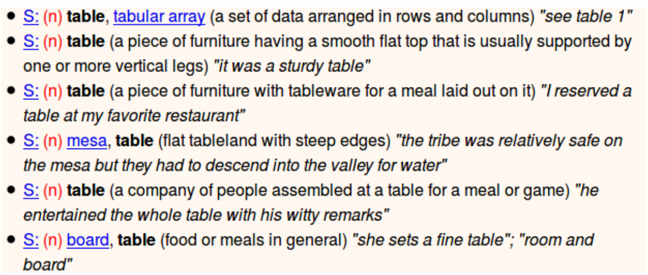
Much work has been done to create strong models for NLI and we show some successful strategies in Section §3.3. Relevant datasets for NLI are introduced in Section §3.2. Before the excessive usage of neural networks, many models heavily relied on external resources, that have either been manually created in order to improve tools for NLP, or arised in a crowd sourced manner 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 solely on distributed word-representations as external information and perform quite good, prior work (Bos and Markert, 2005; Tatu and Moldovan, 2005) depended to large degree on those resources. In Section§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 helpful for improving the performance on NLI. Research has shown that both, manually created and crowd-sourced resources, can successfully be applied in many tasks of NLP. In 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 thre 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. Yet, they encode much more fine-grained sense-differences, that differ only slightly from each other, compared with the difference in meaning off the first synsets. While lemmata within the same synset refer to the lexical semantic relation synonymy, other lexical semantic relations like hypernomy, antonomy and holonomy (as described in Section 2.2, however more fine-grained[12] within WordNet) 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 "church" 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

[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-* , *member-* and *substance-holonyms*.

objects. This is a known issue (Resnik, 1995) and strategies have been proposed to reduce the complexity of WordNet, if the specific domain is known, for instance using sense clustering (Prakash et al., 2007).

3.1.2 Wikipedia

While WordNet contains manually annotated lexical relations and is easily and automatically accessable, Wikipedia[13] is a huge multi-lingual, continuously growing encyclopedia, 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). Even though it has not been created for the purpose of serving as a lexical knowledge base, it still may be seen as partially annotated resource, due to artifacts like hyperlinks. These 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:

- **First paragraph:** The first paragraph of an article can be interpreted as the *sense definition*, since every article covers only one aspect due to the nature of encyclopedias.

- **Hyperlinks:** *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:** Hyperlinks between articles can be considered as *sense relations*.

- **Translation Pages:** 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. YAGO mainly targets to contain a large amount of world-knowledge with Wikipedia, as being tremendously larger than WordNet, and additionally contains relations to express facts derived from it. As opposed to YAGO, UBY (Gurevych et al., 2012) aims for lexical semantic richness. In addition to Wikipedia and WordNet, seven other resources are combined together, providing lexical semantic knowledge in German and English. The combination is realized by using so-called *sense axis*, links between 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 available, but for the most part de-coupled from neural approaches, still lacking this exact 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), where they reach state-of-the-art results. Previous NLI tasks like FraCas (Cooper et al., 1996) or the PASCAL challenge (Dagan et al., 2006) only consisted of a very limited amount of training data, such that neural models could not be used successfully. 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. The corpus consists of 570,152 human written sentence pairs and differentiates between the three labels, described in Section §2.1.

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

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.

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 3 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 3: Example sentence pairs, taken from SNLI, showing typical sentences within the dataset.

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

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 is sufficient to only detect one several indicators. 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), have more specific term in the hypothesis as in the premise (neutral) and show 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 to identify that "basketball" and "volleyball" are mutually exclusive, which shows, how co-hyponyms influence the relation label. While sentences are often a bit longer than in this example, the required knowledge, as 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 test set, serving as evaluation 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 (containing the additional genres). 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).

[15] At least in one sense, not in the sense of *being a ball*.

[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 Section §3.3.

Looking into data

Table 4 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*.

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

As the authors followed the same guidelines, as 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).

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 of previous work, including SNLI, 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. **Hypothesis:** Given the short factual answer-candidates and a question, a new sentence was synthesized using the question and answer. This sentence serves as the hypothesis. For instance the question "When waves of two different frequen- cies interfere, *what phenomenon occurs?*" and the orrect answer "beating" is transformed into "When waves of two different frequencies interfere, *beating occurs*" (Khot et al., 2018).

2. **Premise:** 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. **Label:** While hypothesis, derived from an incorrect 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 (the premise). Thus, samples where croud-sourced annotated, to ensure a correct labelling, only keekping those samples as entailment, that were labelled to have *Complete Support*[20].

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

[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).

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 5 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. 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.

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 5: 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, as 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 inference 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 create 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 only remember certain parts of a sentence after reading it, Chen et al. (2017c) model this human behaviour using gated intra-sentence attention, by generating the sentence representation via pooling[22] strategies over the outputs of the encoding biLSTM. The outputs are reweighted using 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% in 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 other elements, it is not fully differentialbly and thus inefficient to

[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).

train. Soft-attention methods on the other hand, are fully differentiable and weight 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. Furthermore, they use distance masks, that reduce the attention weights, 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.

3.3.2 Inter-sentence-attention-based models

Rocktäschel et al. (2015) shows, that models perform significantly better, when looking at both sentences simultaneously in the senteence-encoding step. 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 SNLI, many works follow 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 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 a simple summation over the comparison-vectors from the previous step. Thus, by using this rather basic aggregation method rather than relying on any LSTM-based method, they reduce computational complexity tremendeously. 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 representations, based on the normalized attention weights. They 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. The composed vector is finally fed into a MLP classifier for the prediction. Chen et al. (2017b) report their results using an ensemble of two inplementations with the same base architeccture. One, as described here, relies on a biLSTM, the other focuses more on syntactic features by encoding sentences with a TreeLSTM. While Decomposable Attention and ESIM achieve competitive results on SNLI we conduct experiments using both models in 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. This is, at the time of this writing, the single best performing model on SNLI. In their approach, they map WordNet relations, as defined in Section §2.2, to a real number $r \in \mathbb{R}$ with $0 \leq r \leq 1$, quantifying the relations between word within p and h based on the path length of each relation within WordNet, and represent each word-pair with this additional feature vector. However evven by enrichening the representations with WordNet information, they only outperform models without external information by a small 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 evidence, that the model leverages from WordNet fused relations for the overall improvement in accuracy[25]. In this paper we show that performance on SNLI is not sufficient evidence for the capability of dealing with simple lexical inferences as inferred from WordNet, which suggests that further investigations should be conducted in this direction.

[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.

[25] This is true for the first published version (Chen et al., 2017a). Subsequentially to work presented within this thesis in Section §5, they show indeed that the additional information from WordNet is a key factor within KIM in their updated version (Chen et al., 2018).

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 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. In this setting, the teacher, being a neural network, is constrained by the rules acquired from an external resource, the student, also being a neural network, considers both labels: the true labels and the constrained predictions of the teacher. By simultaneous training both networks influenced by each others predictions, the logical rules are integrated within the networks parameters, weighted by their learned relevance (soft rules rrather than scrict hard rules). Most attempts to incoprorate external information however, do so by enhancing 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 vector representations for words anyway, any imrovement of word-representations can be adapted with very limited effort to most models.

Joint learning of distributional embeddings with external information

Xu et al. (2014) differentiate between *categorical* (attributes of words like the "gender") and *relational* (relations between words, like "child-of", "is-a", e.t.c.) knowledge and train the word-embeddings from scratch, using three objective functions simultaneously. They use skip-gram to encode distributional properties. At the same time, they minimize the distance between words, that share the same category, thus clustering words by their categorical similarities. Third, they represent a relation as a vect or r, and optimize word vectors, 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. Liu et al. (2015) construct enriched embeddings by defining it as 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. Finally they include those contraints into the training process with skip-gram.

Post-processing existing representations

Faruqui et al. (2015) propose a method called *Retrofitting*, 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 further apart in vector space, while trying to keep 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 improved representations

The demand of integrating lexical resources such as WordNet has mainly been targeted by enrichening word-representations, with previously mentioned approaches being just a small selection. The improvement over standard distibutional word-embeddings of most of these approaches however, is either demonstrated 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, by solely provide intrinsic evaluations. Neural networks for higher level tasks like NLI however reach state-of-the-art performances, still relying on standard pre-trained distributional word-representations like GloVe, even though alternatives exist. Premimitary experiments[26] of using enriched embeddings for SNLI have shown no success. We evaluate the possibility of adding enriched embeddings, following the successful idea of Rücklé et al. (2018), that different embeddings encode complementary information, by concatenating different word-representations. However we focus our experiments on the integration of knowledge on a more progressed step of the network, the sentence representation, due to limited reported success on end tasks using richer embeddings, though many of those embeddings exist.

[26] 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 analyse the sentence-representations of Shortcut-Stacked Encoder†, by visualizing how they encode natural language sentences coming from SNLI (Section §4.2) and how they use the created sentence representations (Section §4.3). 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 and decision criteria 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 visualizations of sentence-representations to date focus on attention-based approaches, showing how words are aligned to each other, such as done by Shen et al. (2018) or Im and Cho (2017). To the best of our knowledge, no or little insights have been gained to understand the final real-valued sentence-representation in vector space. In this section we demonstrate, how a sentence-representation, arising from max-pooling, can be interpreted, using the Shortcut-Stacked Encoder† as the model to analyse. 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.

4.2 Insights on the sentence representation

In this section we show, how we analyse and interpret the information, that is present within the sentence-represenions of the proposed model in Section 2.3. We do so by identifying, 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, consisting of natural language 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, the representation r captures the relevant information for the task at hand. Many applications represent natural text of variable length as a fixed real-valued vector without a deeper understanding what each r_j actually encodes. We shed light into the dimension-wise meaning of the sentence-representation by identifying which word is responsible for the actual value of r_j.

Method

For simplicity, we explain the applied method on an example, 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 network sequentially iterates over

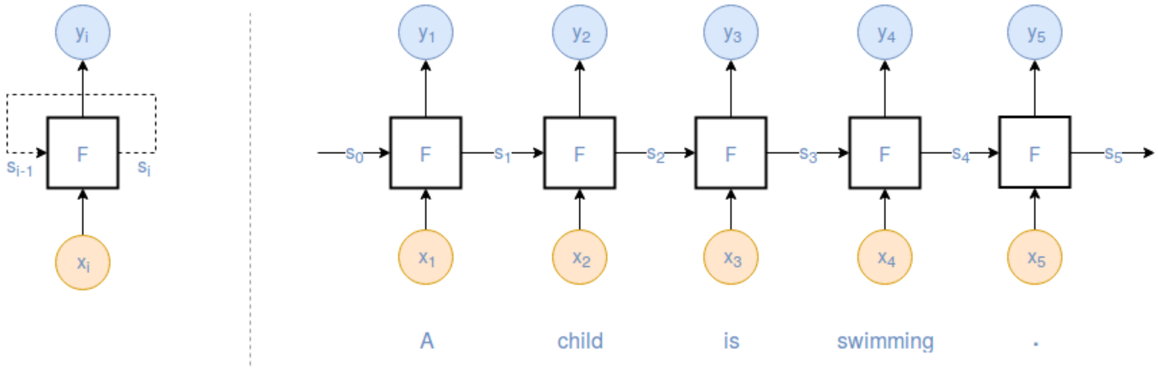


Figure 4: General architecture of a RNN (left). Example sentence in an unrolled RNN (right).

the input sequence x, aggregating in each timestep i the previous state $s_{i-1} \in \mathbb{R}^m$ with the current input x_i, using the function F. This state s_i is then used as input to the next iteration and additionally is output via a mapping function as $y_i \in \mathbb{R}^m$. Multiple implementation variants exist, specifying F and what information is shared across time-steps. LSTMs for instance use several neural gates, to learn what information should be used, should be output or forgotten. This procedure is clarified with an example sentence by unrolling the network in Figure 4 (right). In typical 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 hence contains all relevant information, if optimized for it. Even though the *architecture* of different versions of RNNs is 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 inverting this process. We do so, 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 i a new vector y_i is produced. As done by Nie and Bansal (2017) we concatenate all y_i to a matrix $\mathbb{R}^{m \times i}$, with m being the representation size and each vector y_i being the ith row-vector within M. Assuming a dimensionality of $m = 3$, an possible matrix M, as an example for the given sentence “A child is swimming .”, is displayed in Figure 5. Additionally to creating the sentence representation r by applying row-wise max-pooling on M, we collect the vector a,

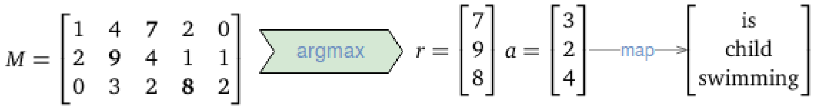

$M = \begin{bmatrix} 1 & 4 & 7 & 2 & 0 \\ 2 & 9 & 4 & 1 & 1 \\ 0 & 3 & 2 & 8 & 2 \end{bmatrix}$
argmax
$r = \begin{bmatrix} 7 \\ 9 \\ 8 \end{bmatrix}$
$a = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}$
map
is
child
swimming

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

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 hence be interpreted by humans. It should be noted, that due to the nature of the multi layer biLSTM each y_i does not only encode the word at x_i but also its context. While this somehow may lead to less accurate mappings or noisy interpretations, we found that the chosen method is sufficient to gain some meaningful insights on sentence encoding.

Analysed data

To reduce noise and for aiming 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 p, and only keep groups, if all samples belonging to the same group are classified correctly. Thus, we reduce the amount of sentences, by removing all samples that are definetly not entirely correct understood by the model, which 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 for each dimension, each sample contains the following information:

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

Lemmatizing, POS-Tagging and dependency parsing were conducted using spaCy[27].

[27] https://spacy.io/

4.2.2 Detection of relevant dimensions

As commonly done, when analysing data, we start by showing a quick overview for the sentence representations at hand. Typically, the Standard Deviation (SD) within a dimension (or any feature in general) correlates with its relevance for decision making. Naturally, a dimension, that does not change its value, and thus being close to constant, is not informative, while a value with a high SD (or variation) 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 of the 2048 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 dimensions 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 (and not captured within the rather small subset of samples), however itserves as a reliable source, what dimensions are relevant to the model and which are not.

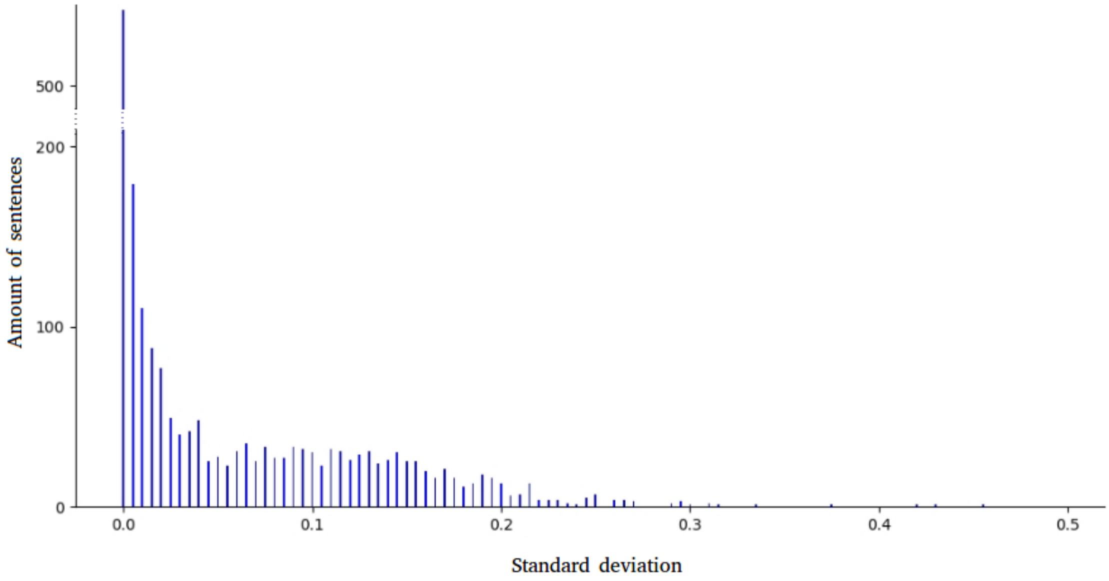


Figure 6: The standard deviation within a dimension of sentence representations (x-axis) by the amount of dimensions with the given standard deviation.

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 same dimension, neglecting the actual value, reached by each word. Especially, for the task of NLI, we assume *semantic*, *syntactic* and *positional* information to be required. Those can all be inferred using the features we extracted 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 reconstruction in original size, but all made of plastic, 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, desiding between those two options. This essentially comes to a major problem when investigating semantic similarities, without prior knowledge, of what attributes may be considered relevant. We therefore investigate the sentence representation, using excessive manual search, in a top down manner: We first search for patterns across many dimensions and many sentences in this section. 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 visualization tool, capable of dynamically generating any labelling scheme for responsible words, based on the features described previously. A sample visualitation is shown in Figure 7. This grid-plot visualizes in each row one particular dimension, listed on the left side as (`<rank in`

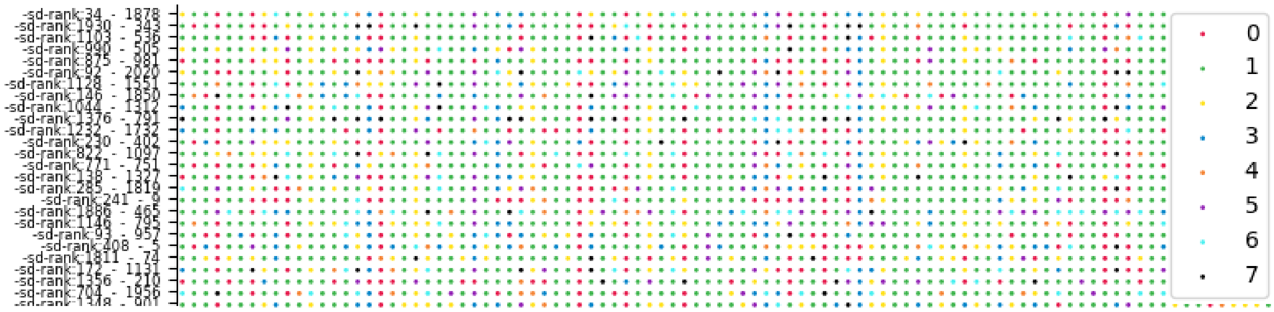


Figure 7: An extraction of a grid-plot, showing dimensions with the position within the sentence of the word, responsible for the dimensional value.

`terms of SD`[28]`> - <dimension index>`). We color the responsible words for each dimension, based on the attributes of interest, shown on the left side. 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 ordered by SD or already pre-sorted by the frequency[29] of a label of interest. In this particular case, dimensions are ordered by their frequency of words on position 1, meaning the dimension within the first row, 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, or colors referring to attributes that may be interpreted similarly. Vertical lines indicate differences across sentences with respect to the attributes of interest and their impact on the visualized dimensions.

Interpretation of positional Information

Word-ordering is crucial with respect to the meaning of a sentence. We evaluate if certain dimensions are aligned to specific word positions, and hence only serve to encode the meaning of the word at a specific position within the sentence. Figure 7 shows dimensions, that are heavily influenced by the second word, indicated by the vast majority of green dots. And indeed, several, also very informative dimensions, are dominated by the second word (ignoring some noise, primarily stemming from other word positions from the beginning of the sentence). Considering the nature of sentences of SNLI, as presented in Section §3.2.1, this is however not enough evidence to conclude, those dimensions correspond (solely) to positional information within the sentence. Taking the merely simple sentence structures (or even only phrases) of SNLI into account, it is very likely that the second word in most cases corresponds to a noun, presumeably describing the main aspect of the image. Optional preceding articles or adjectives may cause this noun to have varying positions between one and three. Looking at the coloring of vertical lines this assumption is backed up, as for each sentence, the responsible word arises fairly consitstently from the same position across most visualized dimensions, indicating, that this stems from the encoded attribute rather than noise. In general we find no meaningful[30] dimensions encoding solely positional information, neither with absolute nor with relative positions, without being correlated strongly with another, more meaningful attribute.

Finding syntactic dimensions

This warrants more investigation using a different labeling scheme, and we look for clues based on POS tags. Tokens are labelled using the Penn Treebank Part-of-Speech Tagset (Marcus et al., 1993) and available in our data with the originally assigned labels. Figure 8 shows an extract of dimensions, labelled by POS tags, pre-sorted, such that dimensions with any single dominant label are shown first. We aggregate different POS-tags, refering to the same concept, together, thus for instance all nouns `NN`, `NNS`, `NNP` and `NNPS` are labelled as `NN`. Several patterns can be seen within this plot. Especially punctuation seems verly well presented at first sight (green and orange). Yet, looking into the actual data and considering their very low SD, these dimensions seem less important. We observe similar issues for dimensions that are dominated by articles (orange). More interestingly are nouns (yellow), being very dominant in diverse dimensions, including dimension `1878` (fourth row), which is also well represented by the second word when checking for positional information (first row in Figure 7). This supports our intuition, that word positions are not directly encoded but merely correlated to other features, like an early noun in this case. Also verbs (blue) are well presented in two dimensions. This

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

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

[30] We do find several dimensions that mostly arise from the first word, yet the resulting value is mostly constant across all sentences and mainly stems from articles like "The" or "A".

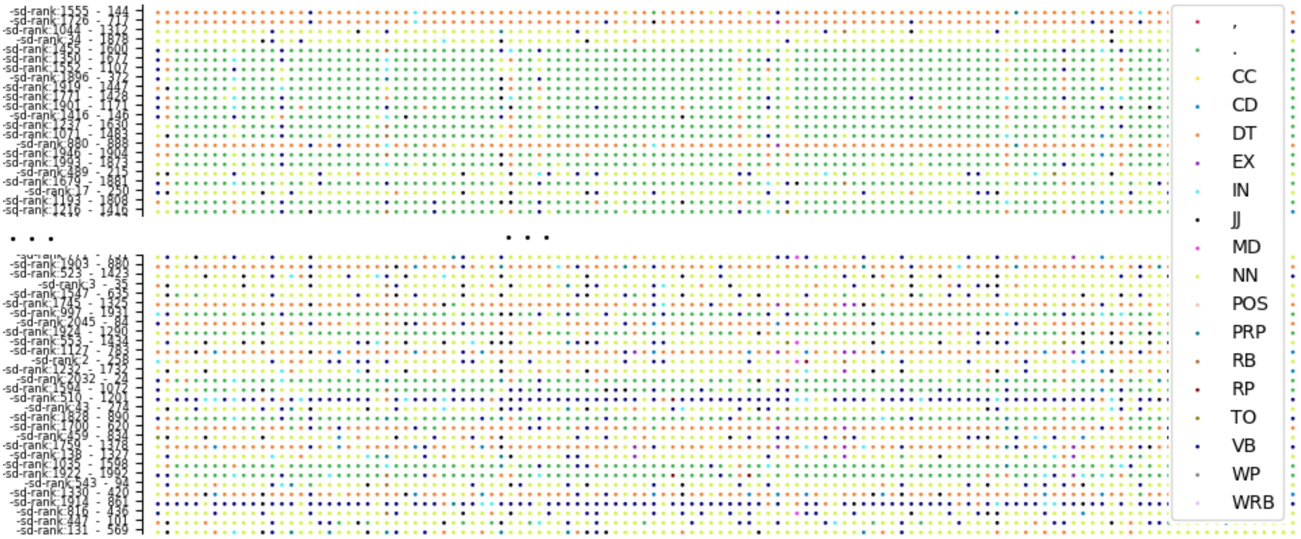


Figure 8: An extraction of a grid-plot, showing syntatical information using the POS tag with pre-sorted rows to have a single dominant label.

plot suggests, that indeed that model captures syntactic information to some extend, represented within the according dimensions respectively, however also shows some drawbacks of our initial naive approach:

1. **Correlation:** As we have shown, different attributes may correlate with each other, thus it is unclear, if a found pattern is a side product of a correlated feature, or the "main thing", being encoded by a dimension.

2. **Non existent information:** Interpreting the meaning of a dimension by the responsible word is only possible if this word indeed characterizes the dimension. In case of positional information we can be certain, that each sentence contains a word that reflects the information with respect to our labeling scheme, as every sentence contains words with all positions. However when looking for encoded information that is not present amongs all sentences, another arbitrary word will still be responsible for the dimension's value. In our case, when finding syntactical clue for instance, most sentences have punctuation, nouns, determiners or verbs. A dimension representing adjectives however will always look very noisy, as it can only be represented by adjectives in the sentence contains such a POS.

3. **Representation value:** We did not yet consider the actual real-valued representation, as used by the model for prediction. Especially considering the previous issue, we expect the model, to have some kind of encoding to differentiate, whether the information of a dimension is present or not.

While we will never completely get rid of the first issue, we try to remedy misinterpretations coming from all three issues by closely analysing dimensions separatley in Section §4.2.3 together with the actual values of r in the dimension at hand. We reduce the shortcomings from the second issue by adding a filtering option, that we demonstarte in the next section, when identifying dimensions containing semantic information.

Finding semtantic dimensions

Looking at the actual data, we observe that several dimensions only include words referring to female humans. We investigate this finding by looking for dimensions that contain gender-specific information. Based on the data we create two wordlists for female[31] and male[32] humans respectively. Following our observation, that more specific information causes more noise in the visualization, due to sentences not containing it, we only consider sentences containing at least one word from at least one wordlist. The resulting plot is depicted in Figure 9 with dimensions sorted by their informativeness according to SD. Words are labelled as *male* or *female*, if they occur in the according wordlist, any other word than those is labelled *OTHER*. Several interesting findings are shown here. The first two dimensions, having the highest variation, do not distinguish between female or male humans, but instead jointly encode both information, seemingly focusing on humans with a specified gender. More importantly however, we observe several dimensions with

[31] Words in the **female** wordlist: *daughter, daughters, female, females, girl, girls, lady, mother, mothers, her, herself, she, sister, sisters, wife, woman, women*

[32] Words in the **male** wordlist: *boy, boys, dude, father, grandfather, guy, guys, he, him, himself, his, husband, male, males, man, men, son, sons*

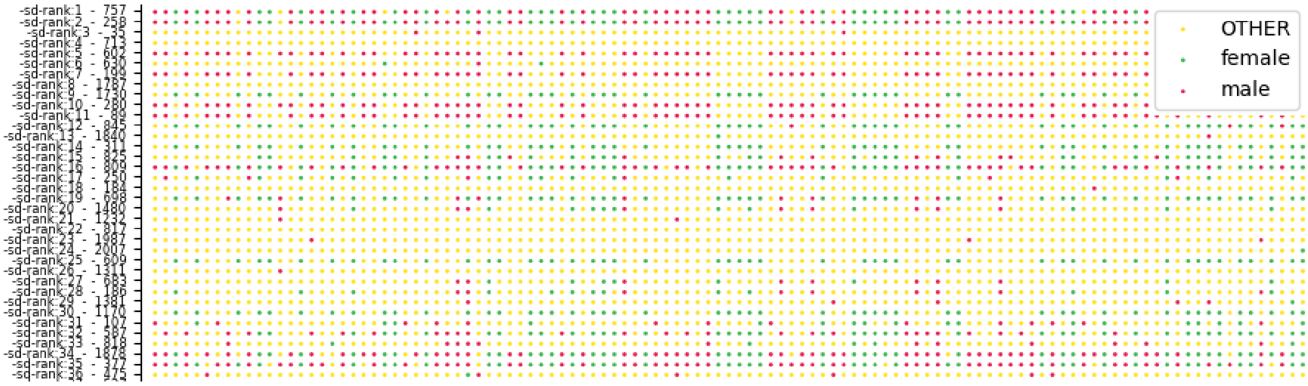


Figure 9: An extraction of a grid-plot, gender specific female using only sentences with words of pre-defined wordlists.

only male (red) and OTHER (yellow) words responsible for its value, while others only arise from female (green) and OTHER only. Comparing these dimensions, we see in the grid plot, that some of them are strongly complementary to each other with respect to the encoded gender: We interpret dimensions that exclusively retrieve their values from female (or OTHER) words, to encode whether there is a female human being present in the sentence (labelled as female) or not (labelled as OTHER). Similarily, dimensions exlusively arising from male (or OTHER) words are likely, to encode whether a male human-being is within the sentence or not. Based on this interpretation, one can observe, that male-encoding dimensions are labelled as OTHER exactly for those sentences, that are labelled as female in female-encoding dimensions and vice versa. Note that all displayed dimensions contain the highest overall SD, indicating that they are amongst the most expressive dimensions of the model. Being redundantly in several informative dimensions encoded, we conclude, that gender-specific information is highly relevant for SNLI. This observation is in line with Gururangan et al. (2018), who show that removing the gender information to create the hypothesis was a common heuristic, applied by the annotators when creating the dataset.

4.2.3 Female and male dimensions

We rely on the method, described above, to manually find patterns, that are encoded in the sentence representations, and identify the corresponding dimensions. Following the drawbacks of our naive approach we conduct our dimension-wise analysis with respect to the actual values of the given dimension, that are retrieved from each word. We represent each dimension in a histogram by mapping the dimensional values from each sentence into a discrete space. In prelimitary attempts tried to automatically divide the one-dimensional feature space of a dimension into meaningful intervals, however this did not create meaningful representations.

Female and male dimensions

Figure 10 shows a detailed view of two of the male-encoding dimensions with no filter applied, thus using all 1000 sentences. For each sentence we retrieve the value of the two displayed dimension and assign it into bins of size 0.05, displayed on the x-axis. The amount of sentences containing a specific value is displayed on the y-axis for each bin, colored by the chosen labelling scheme, in this case, if they are within the chosen word-groups. Based on our previous observation we assume the gender-dimension to only encode whether, a human with the given gender is within the

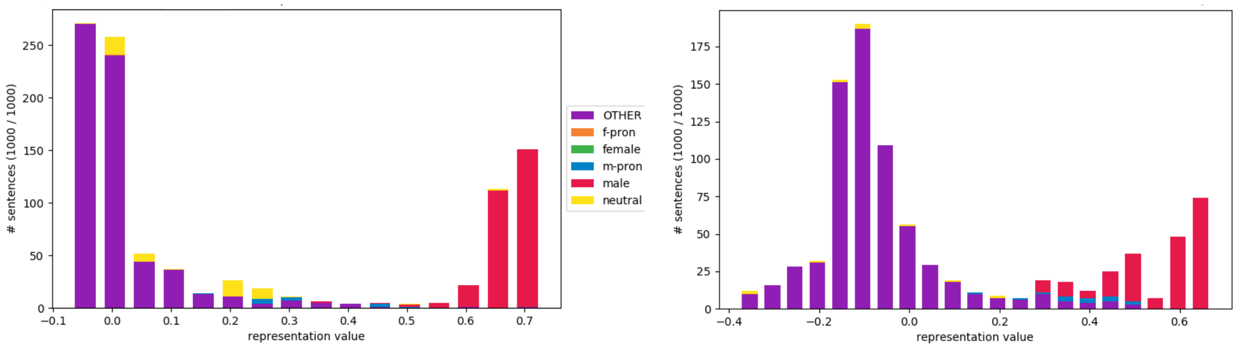


Figure 10: Representation visualitation with respect to genders of dimension 199 (left) and dimension 602 (right).

sentence or not. To distinguish terms for humans without a specified gender from male and female humans, we create third category, containing words for humans without a specified gender [33]. Additionlly we move pronouns from our previous wordlists into new categories. Both dimensions undermine our initial assumption, that they encode whether a male human is present within the sentence or not. Clearly this is seperated by the value within the dimension. All high values arise from male words, most of them covered by our rather limited word-list. Not a single value stems from any word of the female word-list. While neutral words may be responsible for values within these dimensions, their influence is negligable. All words coming from lower-valued bins seem arbitrary, arising from the fact that some sentences do not contain any male words and thus a random word will take its place, resulting in a low value. Even though the distributions are different, both presented dimensions seem to encode the same information. For an even more detailed view, we focus on the individual words from the male wordlist in Figure 11. We observe, that both dimensions encode a fine-grained differentiation between different words and their meanings, even within their high values. In both cases, “boy” scores a lower value than “man” as being *less male*, indicating that this dimension not corresponds to the biological male gender but to attributes, that are generally assoziated with males. This only seems logical, as it is known that gender information is present in distributed word-representations (Mikolov et al., 2013b), that the model relies on. These again, are determined by their surrounding context, which obviously is dominated by male-*assoziated* words. While both dimensions seem to encode the same information, based on the words reaching high values, the encoding of this information slightly differs. Dimension 199 has the tendency to score higher values if multiple males, namely “boys” and “men”, are present, whereas dimension 602 reduces the value for plurals.

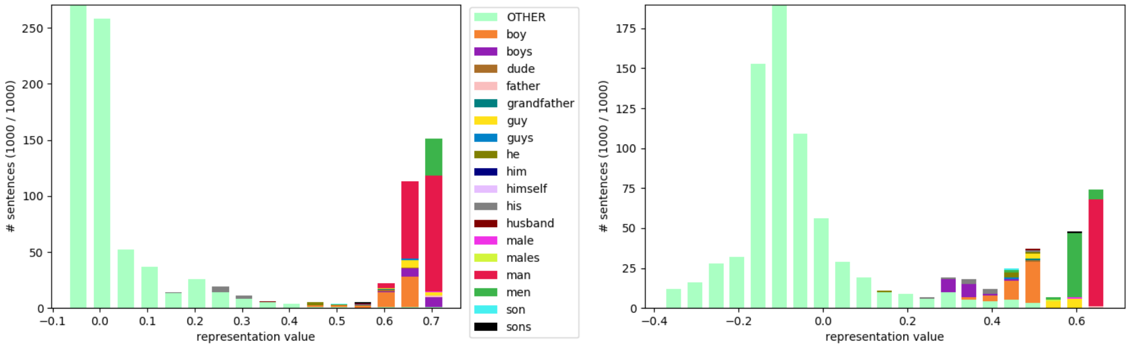


Figure 11: Detailed representation visualitation of different terms for human males of dimension 199 (left) and dimension 602 (right).

We simillarly investigate the female-encoding dimensions, depicted in Figure 12, already using the detailed labelling and observe the same principal encoding scheme. As for the male-encoding dimensions, all higher values within both dimensions arise almost exclusively from our female word-list. Younger females, namely “girl[s]”, are encoded, using a lower value than “woman” or “women”. Furthermore, the different encoding of both female dimensions of singular and plural is aligned with the differences, depicted in the male dimensions. Subsequent visualizations of other dimensions show similar results, such that higher valued words may easily grouped by some attributes, while low values words seem

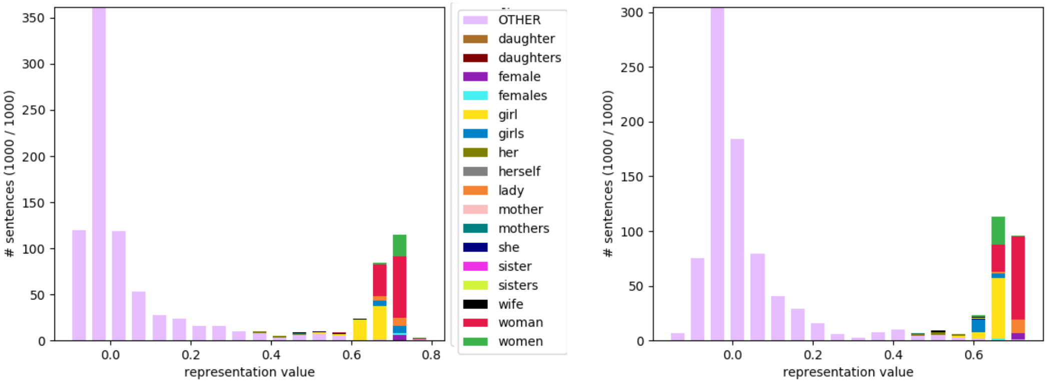


Figure 12: Detailed representation visualitation of different terms for human females of dimension 845 (left) and dimension 311 (right).

[33] Words in the **gender-less** wordlist: *parent, parents, friend, friends, person, people, familiy, student, adult, adults, couple, couples, child, children*

rather arbitrary. We conclude that each (or most) dimensions encode some specific information of any kind. We denote this information of a specific dimension as ξ and use both formulations exchangable within the remainder of this thesis. High values within each dimension are used, if ξ is present within the sentence, and low values, if ξ is not present. Note that this explanation intuitively can be aligned with the max-pooling. Given that ξ is within the sentence, the model adjusts its weights, such that the dimensions, encoding ξ, will result in high values, which naturally will be selected as being the highest amongst all values. In case ξ is not within the sentence, any other arbitrary word will have the highest value of the specific dimension, however this will be significantly lower than in the previous case. We show further evidence for this explanation in the remaining subsections.

Relevance of female and male dimensions

We now closer examine, how relevant those identified dimensions actually are, when the predicting the relations on SNLI. We conduct an experiment with sentence representations, solely consisting of dimensions that we identified to encode gender-specific information. Specifically we found four dimensions to encode each gender respectively. In the first experiment, we only consider subsets of these dimensions as sentence representations, and train a new neural network for each subset, consisting of an equal amount of female and male dimensions [34]. We train the model for 5 iterations using the same hyperparamters as in the Shortcut-Stacked Encoder†. Solely the size of the hidden layer is changed with respect to the size of the sentence representation, due to being tremendously reduced (by only considering male and female dimensions). The results in Table 6 show, that a sentence representation, solely consisting of one dimension per gender, reaches an improvement of about 9 points in accuracy over a random baseline with 33.33%. Adding more dimensions somehow reduced the additional improvements, indicating some redundancy, but also some distinct information is encoded within those dimensions. About half of the data can be classified correctly based on only 8 dimensional sentence-representations, encoding the gender-information of the sentence. This indicates, that these dimensions are highly relevant within the model, however all these observations may also arise from the newly trained model, learning patterns, that are not considered by the original model. We thus conduct another experiment to shed more light into the impact of the detected dimensions for the Shortcut-Stacked Encoder†.

$\|r\|$	MLP size	Acc. (train)	Acc. (dev)
2	6	42.26%	42.87%
4	12	47.11%	47.41%
6	18	47.50%	48.64%
8	24	49.54%	50.37%

Table 6: Accuracies achieved on SNLI using $|r|$-dimensional sentence representations of gender-specific dimensions.

Inverting gender information in the sentence-representation

Knowing, what information is encoded and how this is done, we try to twist the sentence-representations before they are fed into the classifying MLP, hence identify, if it is possible to exploit the gained knowledge for adjusting the represented meaning, without adapting the actual input sentences. Specifically, we try to invert the meaning of the found gender-specific dimensions: If the sentence-representation originally contains information that a male or female human is present, we change it to be not present and vice versa. Let d_j^i denote the value of the ith dimension within the jth sentence representation. For all i, referring to gender-encoding dimensions,[35] we calculate the maximum reached value, denoted as d_{max}^i, and minimum reached value, denoted as d_{min}^i, over all n sentences, whereas n is the amount of sentences in SNLI train data.

$$d_{max}^i = \arg\max_{v^i}(v^i | v^i \in \{d_0^i, d_1^i, \cdots, d_{n-1}^i, d_n^i\}) \tag{6}$$

$$d_{min}^i = \arg\min_{v^i}(v^i | v^i \in \{d_0^i, d_1^i, \cdots, d_{n-1}^i, d_n^i\}) \tag{7}$$

We further calculate the new value $\bar{d}_j^i$, replacing the original value d_j^i, for all relevant i using the following equation. Note that we replace d_j^i by $\bar{d}_j^i$ prior to the feature concatenation, ensuring to overwrite all relevant features. Basically we mirror each value on the dimension's mean, ensuring that the resulting values are within the valid range for the given dimension.

$$\bar{d}_j^i = \frac{d_{max}^i + d_{min}^i}{2} + \left(\frac{d_{max}^i + d_{min}^i}{2} - d_j^i\right) = d_{max}^i + d_{min}^i - d_j^i \tag{8}$$

Rather than using the sample mean from all sentences, which would be heavily influenced by how much the encoded information is represented within the data, we calculate the mean based on the outer values, intending to focus on

[34] We select those dimensions of all sentence-representations from the fully trained Shortcut-Stacked Encoder†

[35] **Male** dimension indizes: 89, 199, 280, 602; **Female** dimension indizes: 311, 609, 845, 1730

the information-encoding aspect. Considering the distributions of the dimensions, having two peaks, either low- or high-valued, we assume this method to be appropriate for our experiment.

Evaluation of inverted gender-dimensions

We use the proposed method for the full SNLI train and dev data and report our results in Table 7, together with the original performance of the used Shortcut-Stacked Encoder[†]. The table shows a range of experiments with an increasing number of dimensions being inverted, on either both sentences or only on the hypothesis. It can be seen, that even after inverting all four dimensions for each gender, the performance only slightly drops, when applied on both sentenes. This indicates that the performed equation (8) indeed is sufficient to invert the encoded information to a high degree, since inverting the gender in p and h simultaneously should not have a large impact on the final prediction[36]. More importantly, when only the hypothesis' representation is changed, we observe, that inverting a single dimension reduces the model's performance only slightly. Increasing amount of inverted dimensions, also increases the negative impact on the overall accuracy. We conclude that this is due to redundant information, most likely coming from dropout.

Inverted dimensions	Inverted sentences	Acc. (train)	Acc. (train) +/-	Acc. (dev)	Acc (dev) +/-
None	None	87.41	0.0	85.31	0.0
1 female, 1 male	premise, hypothesis	87.36	-0.05	85.25	-0.06
2 female, 2 male	premise, hypothesis	87.25	-0.16	85.19	-0.12
3 female, 3 male	premise, hypothesis	87.08	-0.33	84.97	-0.34
4 female, 4 male	premise, hypothesis	86.82	**-0.59**	84.78	**-0.53**
1 female, 1 male	hypothesis	87.05	-0.36	84.73	-0.58
2 female, 2 male	hypothesis	84.76	-2.65	82.29	-3.02
3 female, 3 male	hypothesis	80.23	-7.18	78.28	-7.03
4 female, 4 male	hypothesis	73.20	**-14.21**	71.69	**-13.62**

Table 7: Results in terms of accuracy of inverted gender-specific dimensions on SNLI train and dev set.

Analysis of inverted gender-dimensions

In order to actually see, that sentence-meanings shifted, according to our expectations, namely male individuals should be interpreted as female and vice versa, we analyse the predictions of the data. The results for our observations, when looking for differences between the models prediction using the untouched or inverted (using all eight dimensions, inverting the hypothesis only) sentence-representations, are shown in Table 8. The upper part of the table depicts samples with human main actors having a specified gender, that are correctly classified by the original model. By inverting the mentioned dimensions, we invert the gender-aspect of these actors and subsequently a "woman" in the hypothesis is afterwards encoded similarly to a "man" by the model within the sentence. We observe that the vast majority of samples, containing the same gender in premise and hypothesis, flip the predicted label when being inverted, hence following this interpretation. Some samples without explicit gender information, like "dog" in the lower part of the table, remain with the same label. Especially for human main actors without a specific gender, we observe unexpected

Premise	Hypothesis	Prediction (original)	Prediction (inverted)
A blurry *woman* eating fish.	The *woman* is eating dinner.	neutral	contradiction
A *woman* practicing for tennis.	A *woman* practices tennis	entailment	contradiction
Three *men* sitting behind a building.	Three *men* are sitting.	entailment	contradiction
A *male* in a green jacket points an imaginary shotgun at the sky.	A *woman* in a green jacket pointing an imaginary gun at the sky.	contradiction	entailment
A young *woman* with a ponytail climbs a white stone structure.	A young *man* has a ponytail.	contradiction	entailment
A little *girl* in brown is playing with two hula-hoops.	The person playing with hula-hoops is *male*.	contradiction	entailment
Two dogs standing in the snow.	The dogs are looking in the same direction.	neutral	neutral
Two people dancing outside.	Two people dancing.	entailment	contradiction
A country band is playing.	A group is playing music.	entailment	contradiction

Table 8: Comparison of samples between their predictions based on the original and gender-inverted sentence representations.

[36] We select only 8 out of 2048 dimensions, that are highly representative for the gender-specific meaning. Correlated information however also has an impact on other dimensions, that we left untouched. Thus we don't claim to have inverted the full gender-specific meaning, but a crucial amount of it.

predictions. One possible explanation could be, that since words like “band” or “people” have no specified gender, the inversion of the hypothesis suggests that people of both gender are present in the sentence. Unlike “dog”, those terms are encoded similarly to humans with a specified gender, since both refer to human actors, which yields in according information in other dimensions. Hence, an example regarding a “dog” is less likely to have unwanted side-effects, as the gender presumeably is only considered by the model in conjunction with other dimensions, representing human-beings.

While this experiment intentionally does not improve the accuracy on SNLI, it shows that it is possible to adjust the sentence representation in a meaningful manner, using the gained insights on how information is encoded. The results give evidence, that in the majority of cases, the new meaning corresponds to the initial intention, yet also comes with some minor side-effects, undermining the need to have a deeper understanding how the representations are in fact used by the classifier.

4.2.4 Other semantic dimensions

Following the idea of high valued words being representative for ξ, encoded by a dimension, we analyse more than 100 additional dimensions, finding mostly semantic similarities between the words of interest (high values). Below, we give an overview about the semantic aspects, covered in the representations, and provide sample words, taken from a single dimension each time:

- **Mixed:** The vast majority of dimensions encode even within higher dimensions several different ideas, that can be grasped by looking at the words. For instance one dimension simultaneously considers words as relevant that are related to fast movement or lonely emotion assotiations simlutaneaously (running, runs, race, jogging, no, alone, dry, empty, crying, timid). Another dimension assign all very high values to nouns that are possible arguments to the word *play*, namely instruments and sports, somewhat lower but still very high words reflect assoziated verbs or tools for sporty or artistic activities (soccer, football, baseball, drums, tennis, accordion, guitar, saxophone, dancing, swimmig, sining, painting, boat, bicycle, surfboard). In fact, most dimensions contain words within their high values that may easily be clustered in several groups. Sometimes a single dominant common meaning exists. All examples shown below of course include other words, that may be grouped as an additional category and not necessarily are directly related to the assumed encoded information. However if there is a highly dominant pattern, we ignore words that are divergent to this meaning, considering it as noise. Given the fact, that the representations actually consider the context around the responsible word, we are only able to get an impression of the meaning rather than a accurate defintion anyway, by solely looking at the individual words. Yet, dimensions with several different relations can also indicate, that some dimensions are not individually responsible for a specific ξ. In conjunction with another dimension, those meanings of the words may be seperated from each other, yet we did not inviestigate further into this direction.

- **Community:** Several dimenions encode different communal aspects, like family related topics (children sibling, moter, wife, school) or sozial events (friends, championship, lunch, family, baseball, party)

- **Children:** Different dimension represent children in varying contexts. These dimensions show, that indeed context is captured within the dimensions and they not solely rely on the word, we investigate. Yet, they can be interpreted. Specifically we find several dimensions with children in playful contexts (boy, girl, young, playing, game, teens, skateboarding, soccer) or in the context of caring and comfortness (boy, girl, young, child, sleeping, hungry, tiny, napping, sad, asleep).

- **Locations:** A huge amount of dimension encodes locational information. This may be for instance city or building related (pool, inside, restaurant, sidewalk, floor, bar, classroom, museum, downtown, building) water oriented (beach, pool, water, lake, river, mud, ocean) or referring to different grounds (street, beach, road, sidewak, grass). Several dimensions show topic related locations, together with possible activities, and are harder to categorize (street, beach, outside, park, soccer, rock, truck, boat).

- **General atmosphere** Some dimensions consist of a broad range of words, however it still is obvious, that there is a higher common meaning. One dimension for instance ranges from activities to locations or food, all seemingly indicate some level of “lazy comfortableness” (sitting, inside, sleeping, bed, room, dinner, cream, milkshakes, doll).

- **Activities** Several dimensions include some kind of activity related words, consisting of both, verbs and nouns, clearly showing common attributes (ball, game, race, competition, skateboard, concert, artist), or encoding verbs that usually take positional arguments (walking, sitting, running, standing, walks, jumping), while others solely focus on only one of these meanings (standing, stand, stands, feet).

- **Others:** We found more dimensions, not fitting into a broader category, but still encoding very specific information. For instant one dimension considers everything that has to to with “wearing clothes” as a high value (wearing, dressed, covered, shirt, umbrellas, naked, jersey, dress). Another dimension clearly consists of terms, describing humans using their profession or activity (player, vendor, skier, musicians, clown, workers, jockeys, artist).

While it usually is hard to specifically name the attribute, that most of the high valued words within a dimensions have in common, it is usually very straightforward to grasp the general idea. All these words give valuable information, that enables us to interpret sentence-representations, a large advantage considering that initially nothing was known. Almost exclusively we found these common ideas to be based on the semantic level.

4.2.5 Syntactic dimensions

After extensively looking for semantic information in the sentence-representation, we investigate how much syntax is encoded by looking looking for POS and dependency parse tags.

Verbs and adjectives

We identify syntactic patterns across the sentences, looking for verbs and adjectives using POS tags. One dimension, that is highly dominated by verbs, is depicted in Figure 13 (left). Looking at the actual words, that are responsible for the high values within the dimension however, we find that there is also a semantic commonality between them, with verbs mostly being *playing, walking, sitting, running, standing, swimming*. Several interpretations on what they have in common are plausible, like all taking locations or places as arguments[37] or all being some kind of physical activity[38]. Nouns, scoring high values, exclusively denote sport types like *football, basketball, tennis, baseball, volleyball, hockey*, which represent in combination with a verb also a physical activity. Looking at the words of the adjective-encoding dimension, depicted in Figure 13 (right), we observe an even more obvious semantic relation, since all adjectives[39] are typically used to describe people, even though many different adjectives do exist in SNLI. We face the same problem as described earlier, that different attributes correlate (in this case semantic and syntactic attributes), making a definite interpretation hard. We also observe, that this dimension, like some others, differs strongly with the gender-specific dimensions in their distributions. Gender-specific dimensions consist of two clear peaks, intuitively because they basically encode two states: Either the gender is present or not. This dimension however is encoded using a relatively large range of values, all being relatively equally represented. Event though we do not go deeper into a fine-grained analysis of dimensional values, due to the impact arising from encoded contexts, we assume that other information, as in this case, can be scaled.

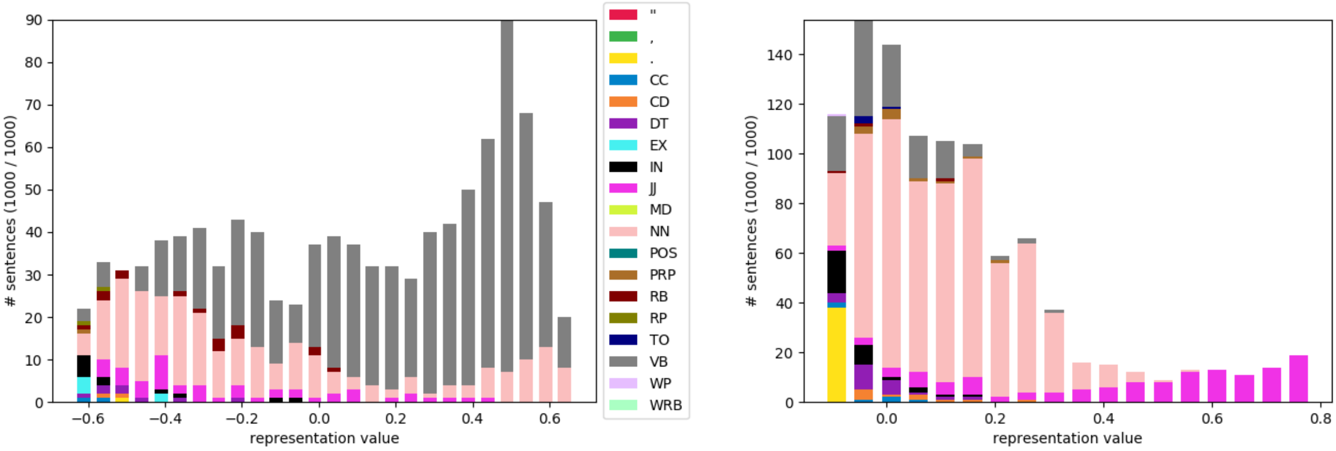


Figure 13: Dimension 713 encoding verbs (left) and dimension 2020 encoding adjectives.

Subjects and objects

The differentiation between subject and object, that can be identified using dependency parsing, seems highly useful for classifying image captions. While the subject most likely refers to the main object, depicted in an image, the object may serve as a more informative explanatio, however most likely being less relevant. To identify whether the model

[37] Especilly considering that there is a large amount of other verbs present within SNLI.

[38] Having different scales how physically intensive it is, in the sense of being sportif.

[39] High valued words of the **adjective** dimension: young, little, old, small, older, fat, large, elderly, lean, middle-aged, ...

learns equivalent information, we look for subjects and objects respectively. We also look for predicates, however this is especially noisy, since many sentences in SNLI actually are noun phrases and thus lack a main verb. The closest, to encoding this information, even by using dependency parsing labels, was the dimension in Figure 13. We find the two dimensions in Figure 14 for both, subject and object, respectively. Similarily, as when analysing the verbal dimension, the first sight suggests that indeed information, as from a parse tree, is encoded within those dimensions. While this actually may be true, undeniably both dimensions retrieve high values from words, that are also semantically highly related. Those of the dimension, seemingly encoding subjects (left), exlucively consist of words referring to people[40]. While they all are very likely to encode a very importnat aspect and also are the subject of the sentence, other subjects, refering to anything else than humans, are not considerd by this dimension. Hence, it seems more likely that the semantic relationship is encoded and just happens to correlate the subject. The object-encoding dimension (right) shows the same phenomen, solely encoding words referring to places[41] with a high value.

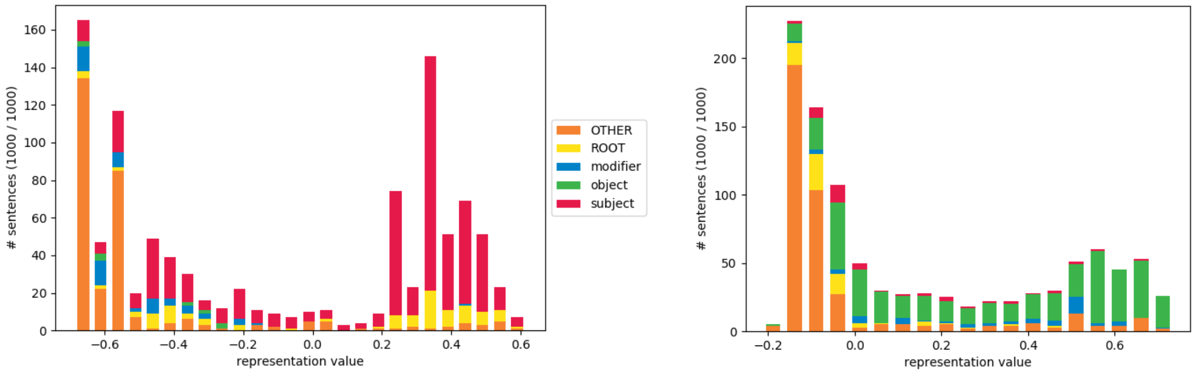


Figure 14: Dimension 757, encoding the subjects (left), and dimension 1840 encoding objects (right) of sentences.

We clearly see that syntactic information is indeed encoded, both for the dependency parse information as well as POS. Yet, those dimensions highly correlate on the semantic level and it seems much more likely that the identification of these semantic patterns is sufficient for the model, to rely on it, encoding syntactic information. This obviously is coming from SNLI, with the majority of sentences regarding people. Another plausible interpretation is that the model does not actually require any syntactical knowledge for the task, based on the simple sentence structure. We conclude that whether the model indirectly uses syntactic information, originating from semantic features, or solely leverages semantic information, not relying on syntax at all, is matter of the perspective, the truth lies probably somewhere in between.

4.3 Insights on the sentence alignment

We have shown, that dimensions represent a specific information ξ of any kind. High values within these dimensions indicate, that this ξ, is present while low values indicate it is not present in the sentence. In this section we analyse, using the newly gained insights, how the models finally aligns the encoded information in the sentence-represenations to predict the entailment relation label. For our analysis we sample 150 premises, each with one hypothesis for each label respectively, that are all classified correctly by Shortcut-Stacked Encoder[†].

4.3.1 Alignment analysis on a single sample

We first analyse single samples in order to identify plausible strategies for the network, when mapping both sentence-representations, initially knowing only very little, how the network actually leverages from the information. We demonstrate our results with the samples premise:

Premise: A woman sitting in the dirt.

and its three hypothesis, one for each label:

Entailment: There is a woman sitting *outside*.
Neutral: A *dirty* woman sitting in the dirt.
Contradiction: A woman *standing* in the sand.

[40] High valued words of the **subject** dimension: man, woman, girl, boy, men, women, boys, girls

[41] High valued words of the **object** dimension: street, beach, pool, outside, park, road, restaurant, sidewalk, grass, city, ...

We highlight the words within each hypotheis, that we consider relevant for the correct label, based on our human judgement. The entailing hypothesis describes, for the most part, the same setting as the premise. Since "dirt" usually appears "outside"[42], we consider this change of words as a generalitazion, meaning "outside" includes "dirt" (amongst others) in this context. The neutral hypothesis introduces new information about the woman being "dirty", which is very plausible, yet not explicitely given in the premise. The contradicting hypothesis clearly is incompatible with the premise, as the woman can either be "standing" or "sitting", but not both. In this subsection we show, that indeed, the identified differences can be easily observed when looking at both sentence-representations simultaneously, for the entailing and contradicting hypothesis. For the sake of brevity we omit the analysis of the neutral sample in this subsection, as it does not give any additional insights.

Visualizing the entailment relation

We start by visualizing the relation between the premise and the entailing hypothesis. We assume that the model will rely on the the information ξ encoded within a given dimension to compare the meanings of two sentences and infer its relation. Figure 15 (left) shows the alignment between between the premise (y-axis) and the hypothesis (x-axis) by counting all dimensions d_i, wheras i is defined by enumerating all dimensions, arising from each word. For each word, we identify all dimensions d_i that are represented by the word in the respresentation. The intersection of a word from the premise and a word of the hypothesis is the total amount of all dimension with the same identifier i, that both words represent. Thus, for instance, "dirt" and "outside" have 176 dimensions in common. This plot is quite noisy, since it does not differentiate between different values, thus even dimensions that encode information which is not given in both sentences are arbitrarily aligned between two words. Following our insights from the previous section, we filter out all dimensions that do not at least have a value of 0.2 in both sentence representations in Figure 15 (right), presumeably resulting in only compared ξ that is present in both sentences. It can be observed that by applying this filtering the main aspects of both sentences ("woman/woman", "sitting/sitting", "dirt/outside") are stongly aligned with each other, while the remaining words only show little similarities w.r.t. their encoding. This incidates, that aligning both sentences intuitively can result in the entailment label within the examined example.

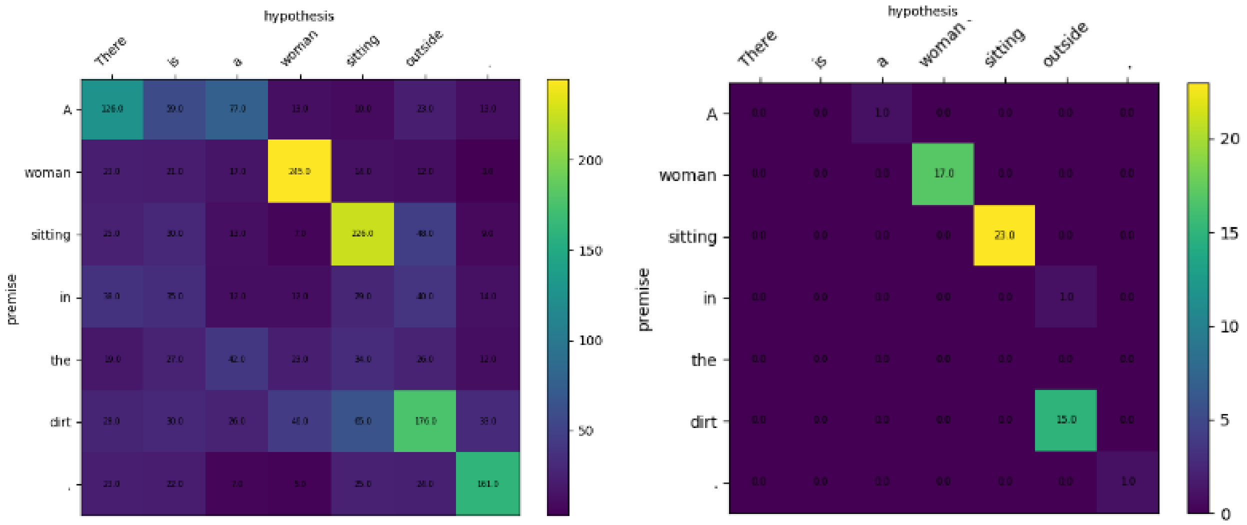


Figure 15: Word alignments of an entailing sentence pair either by counting all shared dimensions (left) or only dimensions with at least a value of 0.2 (right).

Note that the actual Shortcut-Stacked Encoder does not only rely on the original sentence representations only, but also combines them using element-wise multiplication and difference. While element-wise difference intuitively serves a direct comparison of the encoded information per dimension, the effect of the multiplication feature is less obvious. Figure 16 (left) visualizes the mean product of both sentence representations using element-wise multiplication as described in Section §2.3, averaged by the amount of shared dimensions as counted in Figure 15 (left). One can observe that the plot, arising from element-wise multiplications, similarily highllights the same relevant word relations as seen in the previous plot. Showing that similar information is present in both sentences, this indicates that it serves as some kind of soft `AND`-operator. As this plot again might be heavily influenced by irrelevant relations we also visualize the *highest*

[42] Based on its context in conjuntion with "sitting", "dirt" is most likely used in the sense of being a dirty outside ground.

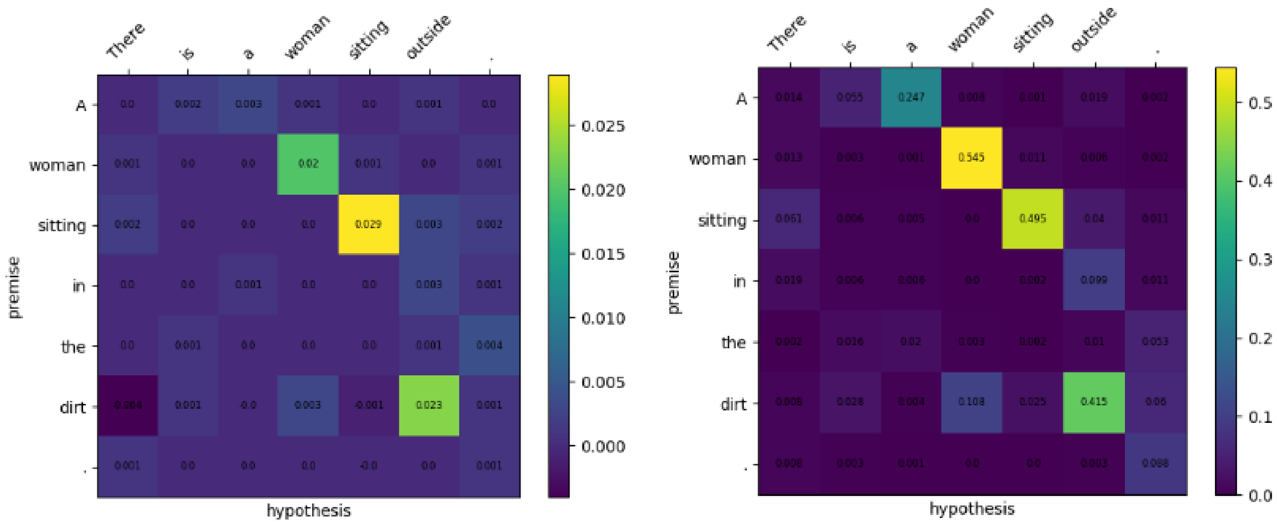


Figure 16: Visualitation of an entailing sample with applied element-wise multiplication either using the mean (left) or maximum (right) product of all shared dimensions for each word pair.

elementwise product, as opposed to the mean, of all shared dimensions between two words in Figure 16 (right), showing a similar pattern.

Visualizing the contradicing relation

We visualize the shared dimensions of both contradicting sentences in Figure 17 (left) in the same manner, as done for the entailment sample, by only counting dimensions that achieve a value of at least 0.2 for both words of each word-pair. Note, that not only the entailing word-pairs show similar values within several dimensions, but also "sitting" and "standing" share the same meaning in some cases. This obviously makes sense, as both verbs are similar w.r.t. several aspects. Since the Shortcut-Stacked Encoder creates sentence representations without looking at the other sentence (without inter-sentence attention) it must encode *all* information, that might be relevant, and is not able to focus on specific relations, that would be crucial for this particular sentence pair. Thus, we also count dimensions, that are distinct between two word pairs, depicted in Figure 17 (right). This is the case for the majority of cases, especially since commpletely unrelated words are encoded by a high amount of distinct dimensions. In order to remove noise, coming from this issue, we apply two thresholds in our visualization:

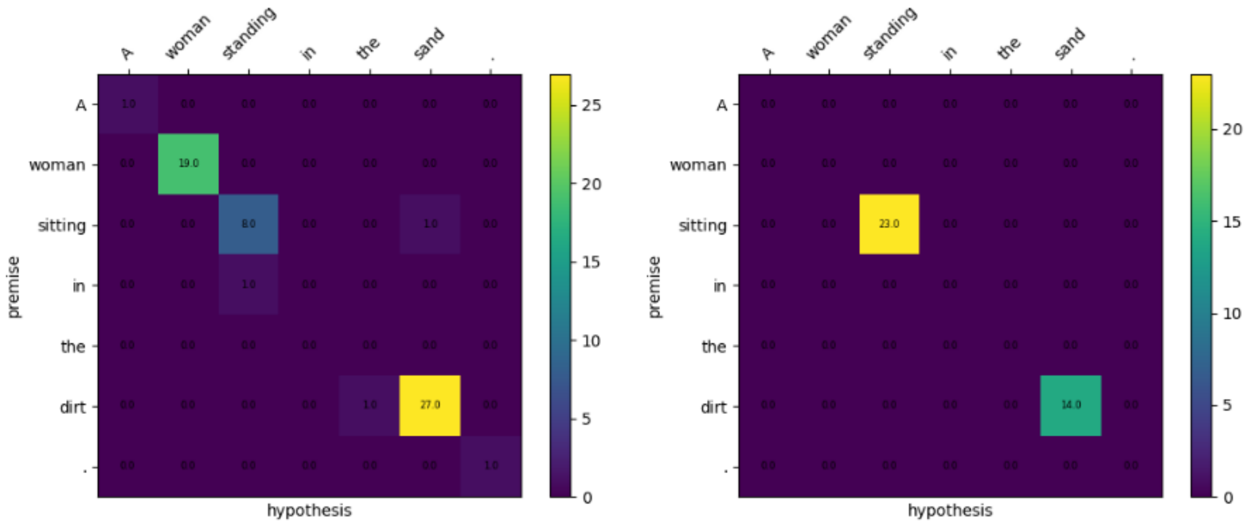


Figure 17: Visualitation of a contradicting sample by counting meaningful shared dimensions (left) and meaningful distinct dimensions 8right) amongst pairs of words.

- **Ensure meaningful relations:** To exclude the counts of dimension between unrelated words, we only consider word-pairs with at least 5 meaningful (in the sense of both values reaching at least 0.2) dimensions. This is motivated by the assumtion, that the model needs to learn which dimensions can be aligned in a meaningful way, which only is plausible if both words also share at least some commonalities.

- **Ensure meaningful value:** Only considering word-pairs with meaningful relations, we only count dimensions that are distinct for both words if the word of interest reaches at least a value of 0.2, thus encodes the presence of ξ.

We observe, that indeed "sitting" and "standing" are encoded using a lot more different dimensions than shared dimensions, and take a closer look at those in Figure 18. The plot shows all "meaningful" dimensions, that only encode ξ coming from *one* of both words. Each dimension show two bars, the left bar indicates the value within the representation of p, the right bar of h. Colors give information about the word, which is responsible for the value. Additionally we leverage from the labelled dimensions, gained for some dimensions in the previous section, by providing sample words of the general meaning encoded by eaach dimension. We only show these indicators for dimensions that we labelled prior to the visualitation, to not be biased towards a specific interpretation. We observe that the identified dimensions indeed highly differ in their value serving as possible features to detect contradiction. This plot also is in line with previous conclusions, that not-given ξ results in low values, coming from arbitrary words. We also see, that knowing ξ (as provided by the labels), is helpful for understanding the shown values, since the labelled dimensions correspond to the responsible word, if having a high value.

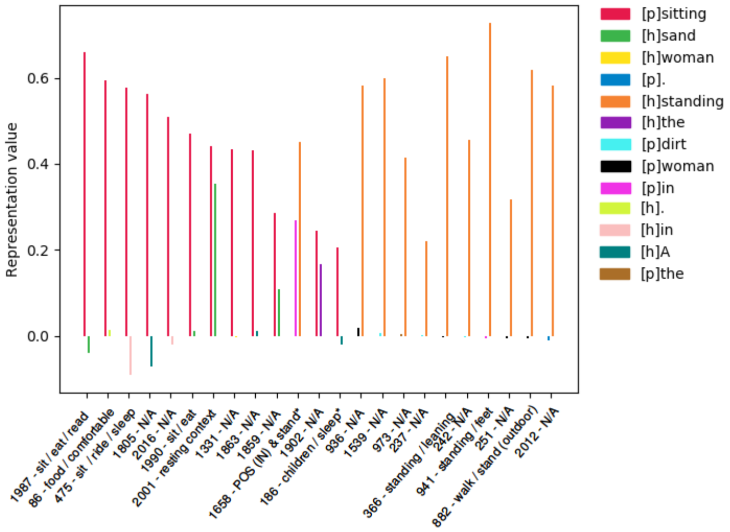


Figure 18: Dimension-wise visualitation of distinct information represented by *sitting* in the premise and *standing* in the hypothesis.

4.3.2 Approach for a general alignment understanding

Our previous results show that it theoretically is possible for the model, to align relevant dimensions an thus infer the entailment label. Doing so, it could differentiate between contradicting or entailing meanings of two sentences. Other than the fact, that the model predicted the examined sentence correctly however, our findings are based on a single sample and show how the model *could*, not actually *does* leverage these information. To also take into account the actual prediction based on the MLP, we conduct another experiment. Again following our conclusions that high values indicate the presence of ξ, we formulate a very simple form of lexical entailment w.r.t. to our identified encoding schemes. Let I_p and I_h be the sets of all ξ, that are present within p and h respectively. We further assume that lower values within a dimension generally represent less information w.r.t. ξ, the information encoded by the dimension:

1. The hypothesis contains a subset of information ($I_h \subseteq I_p$) would either result in paraphrasing ($I_h \equiv I_p$) or in less specific information in I_h, consequently being more general. We expect both cases to be labelled as entailment. We assume that for instance a hyponym "monkey" of the hypernym "animal" contains the same high dimensions as its hypernym and additionally more information that is specific for being a "monkey".

2. The hypothesis contains a true superset of information ($I_p \subset I_h$) results in the opposite case of the one above. The additional information $I_h \setminus I_p$ is possibly true based on the premise, yet not given. We expect this case to be labelled as neutral.

3. The hypothesis and premise contain different information ($I_p \nsubseteq I_h \wedge I_h \nsubseteq I_p$). We expect the model to predict contradiction if the amount of exclusive information in I_p and I_h is relatively large.

In the following, we will refer to those assumpion using their numbering (1), (2) or (3) respectively. First however, we demonstrate this intuition based on an artifically created sample:

Premise: A *green* man is running on the street.
Hypothesis: A man is running on the street.

This is predicted as entailment, as expected by assumption (1) by our model. After swapping the premise with the hypothesis, the predicted label is neutral, as expected by assumption (2). We visualize both sentence-representations in Figure 19 and include additional hints to explain how this visualzation can be read, as we use will use the same technique when looking at multiple samples simultaneously. Intending the figure to serve the validation of our claims, we create it in the following manner: Let $D = \{i \in \mathbb{N} | 1 \leq i \leq n\}$ denote the set of all n dimensions. Furthermore let p_i and h_i denote the value within the ith dimension within p and h respectively. We divide the value range of all values within $\{p_i | i \in D\}$ and $\{h_i | i \in D\}$ into a discrete space, using bins of size 0.1, displayed at the y-axis for p and the x-axis for h with their lower bounds. For each $i \in D$ we identify the corresponding bins, based on the values p_i and h_i and increment the intersecting field by one. Thus, for instance, 1171 dimensions have a value $-0.0241 \leq p_i < 0.0759$ for the premise, while also having a value $-0.0244 \leq h_i < 0.0756$ within the hypothesis. Following our insights, that low valued dimension encode the absence of ξ, we consider the uper left corner as irrelevant for the classification of the relation of both sentences. Arising from the same observations, the diagonale, marked by the red recangle, corresponds to ξ that is present in both, p and h. Subsequently, everything that is above this diagonal represents ξ, that only is present within h and accordingly, everything to the left of the diagonale is only present within p. We investigate the origin of all p_i to the left of the diagonal and observe, that they exclusively emerge from the word "green", which is the only additional information, given in p. Knowing that this naive assumption does not completely hold, we evaluate it on the chosen 450 correctly classified examples. While we find evidence for (1) and (2), we will show why (3) is not sufficient.

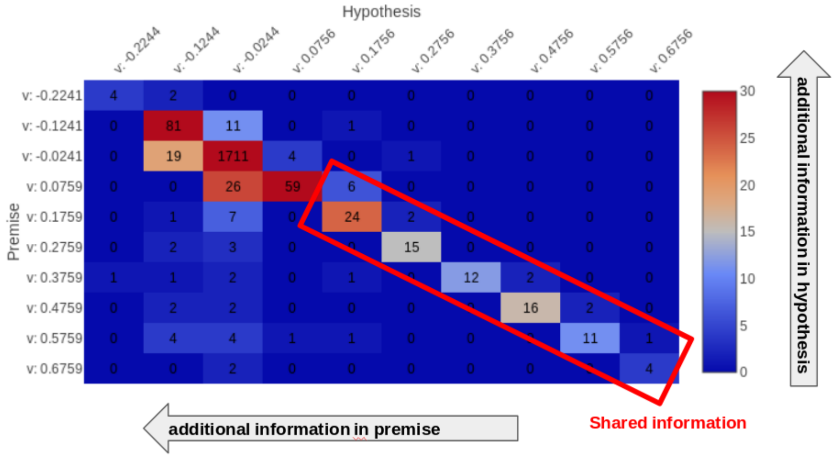


Figure 19: Visualitation of a sample sentence pair with explanatory guides for interpretation.

4.3.3 Entailment analysis

We conduct the same experiment, as conducted using a single sample in Section §4.3.2, over 150 correctly classified samples with the gold label *entailment*. The resulting plot in Figure 20 is calcluated identically to the plot with thee sample sentences, however displaying the mean amount over all sentence-pairs, rather than the the absolute amount. We observe that indeed, the majority of sentence-pairs contains more information within p than in h, undermining our assumption (1). On the other hand only very few information are present in h and not in p. Yet, in order to get a better understanding, we seperate entailment relations based on paraphrasing from entailment based on generalitzation, by re-predicting the sentence-pairs (p, h) with premise and hypothesis swapped, as (h, p). We expect all sentence-pairs, that are predicted *entailment* for (p, h) and (h, p) to be paraphrasing. Accordingly, we consider all entailing sentence-pairs, that are re-predicted as *neutral* after swapping, to arise from generalitation, with the more general sentence h now being the premise. The resulting label distribution after swapping, based on the prediction of the Shortcut-Stacked Encoder†, is listed below:

- **Entailment:** 11 samples (7.3%)

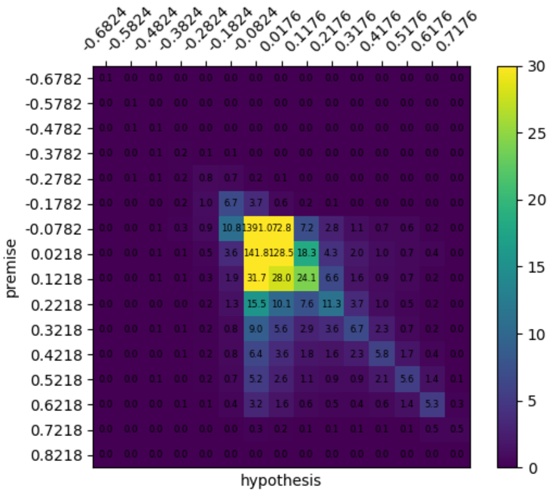


Figure 20: Visualization of 150 sentence pairs (p, h), correctly labelled as entailment.

- **Neutral:** 111 samples (74.0%)
- **Contradiction:** 28 samples (17.7%)

Based on the model's prediction, the majority of cases are described by the second scenario. Only very few samples show the same encoding in both sentences, resulting in entailment. As the re-prediction to contradiction is rather counter-intuitive, we take a look at the actual data. As it turns out, in addition to obvious misclassifications, many of the samples contain quite specific p with a highly general p, for example (before swapping):

Premise:	A girl reaching down into the water while standing at the edge of a river.
Hypothesis:	The girl is outside.

This should definetly falls into the case of our assumption (2), and be classified as neutral, as even after swapping, the original specific p still shows one potential scenario, given the original h. We assume, this arises from the creating process of SNLI, as not many annotators seem to have added a lot more information, when creating the neutral hypothesis. Hence, due to the lackof similar neutral samples in the training, the model does not predict according to our assumption (2), if a huge amount of new information is added. Looking at the data that is re-predicted after spwapping as entailment or neutral, it seems, that those samples are in line with our assumptions (1), (2). We ignore the misclassifications resulting in contradiction and focus on those two labels instead, since they describe the scenarios we initally intended. In Figure 21 we visualize samples predicted as entailment (left) and neutral (right) after swapping. Indeed, samples that seemingly

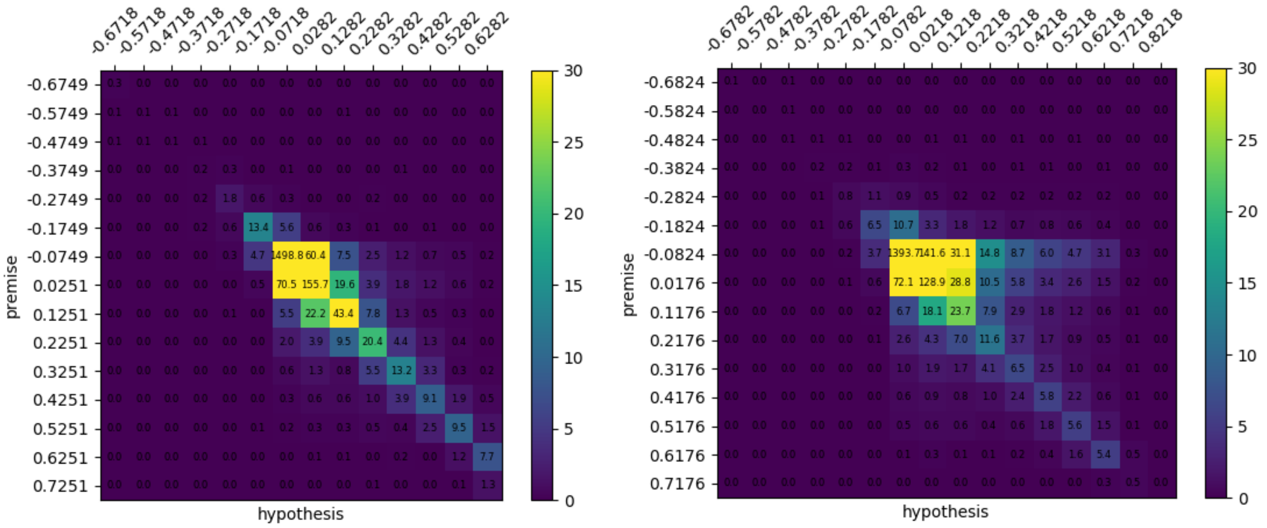


Figure 21: Visualization of samples predicted as entailment (left) and neutral (right) after swapping p and h.

are paraphrased, based on the model's prediction, show highly identical meaning representations over all dimensions. Similarily, all samples that are now labelled as neutral and before swapping were considered entailment, show a hight tendency of encoding only a subset of information of the new hypothesis within the new premise. Both plots show, that our assumptions (1) and (2) are correct, accepting the fact that other scenarios exist, as shown by some swapped contradicting samples.

4.3.4 Neutral and contradiction analysis

We have shown, that we indeed are able to observe, how the model identifies the entailing label, and how this differs from neutral, which in fact follows a very intuitive pattern. We also try to get more insights how neutral sentence-pairs differ from contradiction based on the sentence representations, especially, w.r.t. assumtion (3). The results (without swapping) for 150 correctly classified neutral and contradicting examples respectively are displayed in Figure 22. Unfortunately, we do not likewise find any patterns and also do not succeed by seperating them for a more detailed analysis into smaller subgroups. This does not mean that our assumption (3) is incorrect, since indeed contradicting samples show very distinct information in p and h. Yet the same thing obviously happens for the neutral label. It might be slightly less distinct information when looing at the absolute numbers, however this difference is far from being representative. We find an explanation for this issue by looking into the samples. Consider the following two samples, classified correctly as neutral:

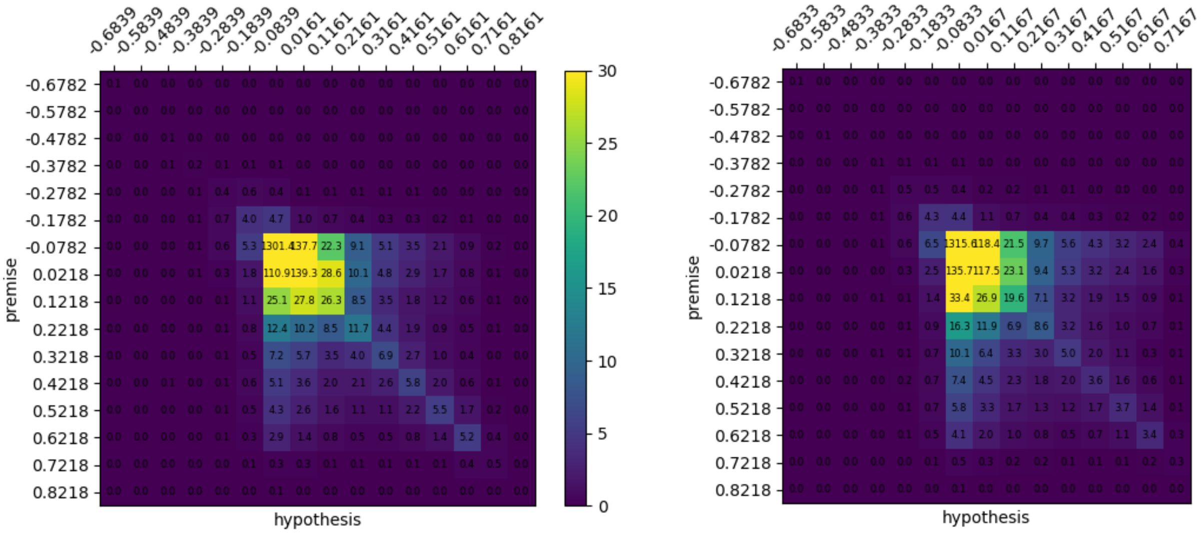


Figure 22: Visualitzazion of 150 sentence pairs (p, h) correctly labelled as *neutral* (left) and *contradiction* (right).

Premise	A group of kite surfers are busy surfing some waves.
Hypothesis	The kite surfers are participating in a race.
Premise	A baby laughing on the floor.
Hypothesis	A baby is being tickled.

In both cases, premise and hypothesis do contain distinct information, yet this information in the case of the neutral sentence-pair is not mutually exclusive but highly compatible with each other, and may also be true. We also look at correctly classified contradicting samples:

Premise	A boy eating at a table.
Hypothesis	A boy coloring at a table.
Premise	A baby laughing on the floor.
Hypothesis	A toddler is crying.

Also in this case, both sentences encode different information, this time however it is not compatible, as the baby is either "laughing" or "crying" and the boy is either "eating" or "coloring", but not both things. We see that assumption (3) holds for contradiction in terms of recall, however it is not sufficient to distinguish between neutral and entailment. We conclude that in order to do so, one must identify, which dimensions can be true simultaneously, indicating neutral, and which dimensions exclude each other, indicating contradiction. Yet, to see how this is done within the model, we need to understand the MLP on top of the sentence representation. This goes back to the main drawback of standard neural networks, being hard to interpret. We found, some useful insights, how the model leverages from the encoded sentence representations, following simple human intuitions. We do not further investigate in the differntiation of those dimensions and end our model understanding analysis at this point.

4.4 Summarizing the insights on max-pooled sentence-representations

We have shown that it is possible, to identify the general meaning of a dimension based on the words, that are responsible for the dimensions value. We observe that high values within dimensions encode the presence of information, which intuitively goes with the nature of the max-pooling mechanism, while low values indicate the absence of information. In our experiment, we additionally have shown, that knowing the encoding-scheme and information of a dimension, it is possible to change sentence-representations in meaningful and intended way, yet observed minor side-effects. In the second section, we showed that by aligning dimensions and knowing their encoded information, it is possible to understand sentence representations well enough, to interpret criteria, leaing to the prediction. In the last part we analysed, how the model actually performs the alignment between the sentence representations, and observed an intuitive explanation for several different types, how sentences relate to each other w.r.t. label.

All these results are based on a relatively limited amount of rather small sentences, stemming from SNLI and thus only show experimental results. For a general claim, similar experiments should be conducted on a larger scale, using more samples, containing longer text and using different models (all with max-pooled sentence representation) to see if those claims still holds.

4.5 Identification of missing knowledge

In order to see, what information is not captured by the model, and can be helpful when integrated, we analyse the errors made on SNLI. This analyses uses Shortcut-Stacked Encoder[††], being closer to the reported accuracy by Nie and Bansal (2017).

4.5.1 Approach

Aiming for missing knowledge, and not for a general error analysis, we focus on misclassified samples with gold label contradiction, predicted as entailment and vice versa, as we find that the label neutral was overall not well understood by the SNLI annotators. To simplify the process we sample, according to this constraint, sentence-pairs having a high lexical overlap[43], and look for the knowledge, required by a human to predict the correct label. In total, we look at 196 samples, incorrectly predicted as entailment, and 163 samples, incorrectly predicted as entailment, to identify common categories of missing knowledge. Note, that by pre-filtering the samples this way, our results exclude certain aspects. For instance many contradicting samples have multiple exclusive elements, resulting in a low lexical overlap.

4.5.2 Results

We show our results with the identified categories, incuding sample sentence-pairs for both labels in this section.

Misclassified contradicting samples

Table 9 shows the misclassifications of samples, labelled incorrectly as entailment. We find that most problems arise from words sharing lexical relations, namely antonomy or cohyponomy, for different kind of POS. We opt to show "amounts" as a seperate category, due to its high frequency. While Verb antonyms may also be considered as cohyponyms, we also list them seperately, if they refer to the opposite meaning. Some samples, listed under "structure", require the model to take word-order into consideration, and is mostly represented by semantic role reversal. We assign all samples to "common sense", that require information, that cannot be retrieved using lexical semantic relations and usually need additional information, which is only *implied* by the described entity or activity. Any sentence-pair, where we could not identify the required knowledge, due to not agreeing with the label or due to being highly ungrammatical, or some rare very specific details, are ignored in our results, categorized as "ignored".

Misclassified entailing samples

Analysing the required knowledge for misclassified entailing samples as contradtiong, we identify different categories, depicted in Table 10. Essentially, we find three different ways of how the relation between p and h can be described, and differentiate between lexical- and world-knowldge, being required for the correct prediction. In the example of

[43] Only considering a sentence pair, if at least 50% of the words within the shorter sentence are contained in the other sentence, using a BoW perspective. Casing is ignored.

Problem	Type	Amount	Example
cohyponym	Nouns	29	A *creek* runs through the grassy area. A *lake* appears in the grassy area.
cohyponym	Verbs	32	The boy is *riding* his skateboard. The boy is *carrying* his skateboard with him.
cohyponym	Amounts	33	*Three* people resting on a snowy mountain. *Four* people are on a snowy mountain.
antonym	Adjectives	13	The ground is *covered* in snow. The ground is *visible*.
antonym	Verbs	14	boy *pushing* wagon with two pumpkins in it A boy is *pulling* a wagon with two pumpkins in it.
antonym	Prepositions	17	A man walking *down* stairs. The man is walking *up* the stairs.
structure	All	15	a sheep chases a dog. There is a dog chasing a sheep.
common sense	All	20	Someone in a 3ft swimming pool. A person is in a very large and deep pool.
ignored	*ignored*	23	A man climbing a rock wall. A man climbs the wall.
Total	-	**196**	-

Table 9: Misclassified samples with gold label *contradiction*, predicted as *entailment*.

Problem	Required knowledge	Amount	Example
Paraphrasing	lexical knowledge	16	Two people play *foosball*. Two people are playing *table soccer*.
Paraphrasing (negated opposite)	lexical knowledge	19	A young boy is *sleeping*. A child is *not awake*.
Paraphrasing	world knowledge	20	girl *opening* cosmetics *bottle* The girl is *removing the top* off the *bottle*.
Generalization	lexical knowledge	30	The two *boxers* are females. There are two female *athletes*.
Implication	world knowledge	53	A hockey player makes a shot. A hockey player *is on ice*.
ignored	*ignored*	25	two people sit on a bench. two people sit on sand near water.
Total	-	**163**	-

Table 10: Misclassified samples with gold label *entailment*, predicted as *contradiction*.

paraphrasing, for “foosball” and “table soccer” or “sleeping and “awake”, it is sufficient to detect that they are synonyms or antonyms respectively. The third case requires some actual understanding of the process of “opening” in the context with a “bottle”. While some of these samples may be understood by knowing several meronyms, we consider them to require a deeper conceptual understanding of how thing work, hence world knowledge. The largest group, “implication”, is interesting, as all h contain additional information, not directly given by p. While this case should usually be labelled as neutral, in this case the additional information is automatically implied[44] (even though not textual) by p. This implication also requires other external knowledge than lexical relations.

4.5.3 Conclusions

We observe that especially the classification of contradicting samples can be improved by using lexical relations, which are available in WordNet, as described in Section §3.1.1. Those may partially be also helpful for the problems, identified on the entailing samples. Yet, in this case, world-knowledge seems more relevant, as could be gained from resources like Wikipedia (see Section §3.1.2). While obviously the goal of NLI is, to have proper reasonng capabilities, including dealing with world knowledge, we emphasize the aspect of lexical knowledge, arguing that any insights on how to incporporate this (simpler) information, may later be used to include world-knowledge.

Note that the identified problems with lexical relations, especially *antonomy* and *cohyponomy* in Table 9, refer to the same problem, stemming from the nature of distributed representations. Essentially, they follow the distributional hypothesis, described by Pantel (2005) as:

> *“[...] words that occur in the same contexts tend to have similar meaning.”*(Pantel, 2005)

Considering a sentence like “The president of Italy hopes to get re-elected.”, one can easily conclude that *Italy* is a country based on its context. Even replacing *Italy* by any made-up country would intuitively result in the same conclusion. Subsequently to being represented by their context words, distributed embeddings supposedly have better generalization abilities (LeCun et al., 2015) as modely may rely on the fact, that similar words are represented similarily. However also mutually exclusive co-hyponyms or antonyms often share similar contexts (like most countries could replace “Italy” in the given example), resulting in very similar vector representation despite opposite meanings in one aspect. This is a known problem (Sahlgren, 2008) of distributed word representations and several approaches, as explained in Section §3.4.1, aim to fix these problems in the embedding space.

[44] For the given sample, we assume that Americans see *hockey* in the sense of *ice-hockey*.

5 Additional SNLI test-set

In the previous section we have identified several lexical inferences, that have not been captured by the model and thus mispredicted. However we could only consider a small subset of misclassified samples, since a small lexical overlap would result in multiple interpretations of what the model understands and what not. In fact, even when identifying missing knowledge in our anaysis, we needed to rely on intuition rather than certainity when assigning mislabelled samples into categories. We also take a look about what the model seemingly knows, based on its correct predictions, shown in Table 11. In the first example humans see, that both sentences describe the same scenario with only the main actor changing.

Premise/Hypothesis	Label
A young boy wearing a jacket pushing a hand mower on the grass. A girl is mowing the grass.	*contradiction*
A man is doing a cannon ball into a pool, stadium chairs fill the background. Someone is jumping into water.	*entailment*
A woman testing a comfortable pillow. The woman 's head is in contact with the pillow.	*entailment*

Table 11: Correctly classified examples.

Aiming for NLI and thus NLU, the model should proceed similarily. It cannot be said, whether the model identifies the paraphrasing of "mowing". Yet, we can see that the only required information for the correct label is the gender of the mowing person. This is, as we have seen in Section §4, a very important feature within the model. In the second sentence, an average human knows, that "doing a cannon ball" is a special form of "jumping" (in the water). However, a simple heuristic that h is entailed by p if it describes a more general scenario, would be sufficient for a correct prediction, if the model is able to identify that a "man" is a "person" and "pool" is somehow related to "water". Even the alignment to "jumping" and "ball" may be given in the representation, since the model mixed words in sportif/activity dimensions. Similarily, in the last sentence pair, we intuitively find it hard to believe, the process of "putting the head in contact with the pillow" is known to be implied when "testing" it to the model. Again, it seems more likely that the high overlap (and potentially the semantic relatedness of "head" and "pillow") are causing the correct prediction rather than actual NLU.

5.1 Goal of the new test set

We believe that the high accuracy on SNLI stems from exploiting these simple heuristics, coming from dataset specifc patterns, rather than actually encoding the correct meaning of the sentences. Additionally, while models relying only on external information from distributed word representations achieve strong results, they still depend on the information encoded within these. Especially for mutually exclusive words, that appear in similar contexts, this information alone might not only be insufficient but also misleading. With this motivation, we create a new additional test set (Glockner et al., 2018) for SNLI with adversarial sentence-pairs, that only differ in one aspect, defined by lexical semantic relations. This will help in three aspects:

- We show that even state-of-the-art models fail to capture simple lexical inferences and a high performance on SNLI is not sufficient evidence for a proper NLU, being heavily dependant on dataset specific patterns. This is motivated by Jia and Liang (2017), who find similar issues in the field of reading comprehension.
- Having sentence-pairs, differing in only one specific and known aspect, enables a very accurate estimation, whether the model has enough understanding capabilites for the particular required lexical relation or not, as we exclude any noise.
- Only measuring the capability for lexical inferences, that are available in a variety of lexical resources, like WordNet, we show the need to incorporate such knowledge bases into neural networks and provide a dataset, to measure the effectiveness of these approaches. Note that improvements (even if applied on the underlying NLU) are hard to show on the original SNLI test set, has models already achieve results at the upper possible bound.

Our claim, that state-of-the-art results highly overestimate the actual NLU capabilities of the model, as they rely on patterns within the dataset, is in line with other works, that tackle the problem from different perspectives. (Gururangan et al., 2018) refer to those patterns as "Annotation Artifacts", arising from similar stragies used by the annotators, when creating the hypothesis for each label. As opposed to our approach, they do not create new samples to reduce the impact of these patterns. Instead, they identify samples, that contain enough information solely in their hypothesis, to be classified correctly. After removing these samples, the remaining dataset shows, that state-of-the-art models perform significantly worse. (Dasgupta et al., 2018) focus on the compositional aspect by atomatically generating sentences from SNLI, rearranging noun phrases in any order around the words "`[not] more / less`", thus requiring the model, to consider the sentence structure[45]. Based on their results, they claim that the high performance on SNLI arises from the fact, that word-overlaps or specific single word relations are often sufficient.

5.2 Dataset

We now describe, how we create the new test set and make sure it, that is correct and *fair* w.r.t. the train data in the way, that it does not introduce new information, but only relies on the generalization ability of information, given in the train data. This is important, since we cannot assume a model trained on a specific dataset to perform equally good on different domains (Goldberg, 2017).

5.2.1 Creation of adversarial samples

We derive all new sentence pairs from the original SNLI train set, by replacing selected expressions within a single sentence. Those expressions usually consist of a single word, however in some cases, like "New Zealand", it consists of several words, according to our definition. For the sake of simplicity, in the remainder of this chapter, we generally speak of *words* rather than *expressions*, covering all of our replacements. The original sentence from SNLI is kept as the premise, while the adapted sentence with the replaced word, serves as the hypothesis. We will in the remainder of thes section refer to w_p for the word within p that was replaced by the word w_h in h. Samples from the resulting dataset are shown in Table 12. Note that traditional RTE systems would consider the first example as neutral, since a man may hold both instruments at the same time. However for being conform with the labelling scheme in SNLI, this is considered as contradicting[46], based on the event-coreference assumption and the most dominant aspects of the image being described within the sentence. This was introduced by Bowman et al. (2015) for exactly this purpose of distinguishing between different interpretations and hence reducing ambiguity of different possible labels.

Premise/Hypothesis	**Label**
The man is holding a saxophone The man is holding an electric guitar	contradiction
A little girl is very sad. A little girl is very unhappy.	entailment
A couple drinking wine A couple drinking champagne	neutral

Table 12: Examples from the newly generated test set.

Generation of word-pairs

We manually generate a list of word-pairs (w_p, w_h) from online resources for English Learning [47]. They provide large lists topically clustered words, which we use to derive co-hyponyms, as well collections of as rather generally applicable synonyms or antonyms. We focus partly on entailing, but mostly on contradicing examples, and assume synonyms to refer to the former, co-hyponyms and antonyms to the latter case. Doing so we must consider the following things:

- **Compatible co-hyponyms:** Co-hyponyms not necessarily exclude each other (Kruszewski and Baroni, 2015). A "jongleur" for instance might also be a "clown", even though both could be considered as neighbouring hyponyms of artist, while a "horse" and a "cow", both hyponyms of "animal" may not both refer to the same entity. As the newly generated sentence-pairs naturally will have a high lexical overlap, the prediction may be emphasized towards neutral anyway because if some distinct information. To reduce the impact of this rather random decison criteria, we mostly aim to find out, whether the model is able to identify contradicting examples. Thus, we focus on mutually exclusive rather than compatibe co-hyponyms. Similarily, we remove word-pairs, that commonly are confused by humans like "pink" vs. "purple".

[45] For instance: "The man has more hair than the woman." vs. "The woman has more hair than the man."

[46] We verified this by finding similar samples within the actual SNLI dataset, also labelled as contradiction.

[47] http://www.enchantedlearning.com

- **Polysemy:** In order to automatically generate new sentences from (w_p,w_h), we require both words to have one highly dominant sense, such that both words are generally replacable, leading to correct sentences. We verify this, by sampling random sentences, containing w_p, to see their usage within SNLI is conform with w_h. Words that appear in highly different senses are excluded. The country "Jordan" for instance, is mostly used in the sense of the basketball player *Michael Jordan*, and thus is not used. We observe a similar problem on much more fine-grained level. The word-pair of the antonyms (*old*, *young*) both contradict each other on a very general basis. Yet, whether they can be replaced or not is dependant on the context. While "old" may refer to *things* as well as people (like "an old computer" or "an old man"), "young" usually can only be used in combination with people. We thus distinguish between ($w_p \leftrightarrow w_h$), that can be swapped both ways, and ($w_p \leftarrow w_h$), that only can be swapped in one direction. In this case the more restricted term "young" can be replaced by "old", not vice versa, as we aim for a high precision on correct sentences.

- **Structural word usage:** Furthermore, (w_p, w_h) may be used together with different function words. Consider the for instance ("day", "night") or ("near", "far"). While both (w_p, w_h) represent opposite meanings for a high amount of contexts, replacing one with each other leads to invalid sentences like "John sleeps at day[/night]" or "The house is very near[/far] from the sea". We identify these patterns by looking at the word usage, and extend the (w_p,w_h) adequatly with the function words like ("during the day", "at night"), automatically reducing the chance of incompatible senses as a side-effect.

We manually evaluate all selected (w_p,w_h) for synonyms and antonyms, based on the points mentioned above. Topic related co-hyponyms are only individually evaluated, as mapping each word with each of its co-hyponyms is rather inefficent to manually verify. In addition to that, we create antonym word-pairs from WordNet, sharing the same POS and having a cosine similarity of ≥ 0.5. In total we generate 3990 word-pairs[48] and keep the topical (in case of cohyponyms) or relational (for synonyms and antonyms) information, leading to 13 groups[49]. To ensure that we not confront the models, trained on SNLI, with new information, we verify that each word indeed is within the train-data and the used word-embeddings. In the final test set, frequencies[50] from newly introduced words w_h range from a single occurence (e.g. "Portugal") up to 248,051 occurences ("man") with a mean of 3,663.1 and a median of 149.5 (interquartile range 19.0 - 1107.5). As the general goal of machine learning is not to memorize, we aim for this distribution, having a high amount of less representative words, to measure the models generalization power. We motivate this, as it is very likely in a real-world scenario, to encounter samples requiring this kind of knowledge, even though it may not be omnipresent within the train data, and hence should be inferred from the learned features.

Generation of sentence-pairs

As previously mentioned, we derive all our samples from premises of from the training set. Theses premises serve in the exact same form as the premise of our new samples, while the hypothesis is generated by replacing w_p within p by w_h. Thus, not only are the newly introduced words known from the training process, but also each p has been seen during training in the exact same form, and has been encoded with respect to at least three hypothesis, for each label respectively[51]. By doing so, we intentionally violate common practives in machine learning, as the test set is not completely isolated from the train data, wich should serve as an advantage for the model. We finally remove highly unlikely sentences by ensuring that the bigrams (w^{t-1},w_h^t) and (w_h^t, w^{t+1}) with w^t being the tth word within h must have a frequency of at least 10 in the wikipedia bigram corpus[52]. In case the replacement consists of several words, w_h^t corresponds to the first or last word of this expression respectively. This preprocessing helps to clean the created samples, yet two problems remain.

Remaining issues

Especially on the semantic level, our newly created sentences may still be incorrect. For instance consider the following sentence-pair:

Premise:	The car would not *start* and, consequently, stayed in this garage.
Hypothesis:	The car would not *end* and, consequently, stayed in this garage.

Obviously both bigrams ("not", "end") and ("end", "and") appear quite commonly within a large text corpus. However does "end" neither serve as an appropriate verb for "car", nor would the resulting sentence (even if a more applicable

[48] We count the replacement directions separately, thus ($w_p \leftrightarrow w_h$) counts as ($w_p \rightarrow w_h$) and ($w_p \leftarrow w_h$).

[49] Countries, Nationalities, Colors, Numbers, Antonyms, Synonyms, Vegetables, Drinks, Loction-Verbs, Materials, Planets, Rooms, Instruments

[50] The amount of individual sentences containing w_h in the exact surface form.

[51] Due to the generation process of SNLI.

[52] https://github.com/rmaestre/Wikipedia-Bigram-Open-Datasets

word like “stop” would be used) make any sense to a human.

Furthermore, we need to assign the correct label for the relationship between both sentences. Knowing that our newly created p and h only differ in one word, one could consequently assume that the relationship between p and h is the same the relation between w_p and w_h. Thus for instance, one could label (p,h) as contradiction, if w_p and w_h exclude each other, or as entailment, if they have synonym meanings. This heuristic may indeed be correct for most cases. MacCartney and Manning (2007) however show, that this is only the case for upward-monotone sentences. In upward-monotone sentences, replacing a word (e.g. “cow”) with a more general term, like a hypernym (e.g. “animal”), yields in a broader meaning coverage of the sentence and thus results in entailment. MacCartney and Manning (2007) identify several linguistic patterns, like negation or restrictive quantifiers (e.g. “without”) or verbs (e.g. “fail”), that result in downward-monotone sentences, yielding to different results w.r.t. entailment relation. This differentiation is not only relevant for hypernyms but also for other lexical relations, as shown in Table 13 with the example of co-hyponyms. In both examples “France” is replaced by its co-hyponym “Italy”. Even though both words are mutually exclusive only for the upward-monotone sample, the sentence relation reflects the relation between both words. The same implication does not hold anymore for the downward-monotone sentence. Even though we can assume that most premises in SNLI are upward monotone, especially since image captionas are more likely to explicitly describe the content of the picture, we must take into account, that whether the sentence relations corresponds to the relation of (w_p,w_h) or not, depends on the context of these words.

	Sentences	**Label**
Upward monotone	John is hiking in *France* John is hiking in *Italy*	contradiction
Downward monotone	John is hiking outside of *France* John is hiking outside of *Italy*	neutral

Table 13: Comparison of co-hyponyms in upward-monotone and downward-monone sentences.

5.2.2 Validation

We address both mentioned problems, by annotating the new test set using crowd-sourcing with Amazon Mechanical Turk[53]. In order to make this a cost-effective process, we aim for the Human Intelligence Task (HIT) to be simple for the annotators while at the same time enabling them, to validate and label as many as possible samples. We constrain ourselves, such that one HIT contains five hypothesis, which are all originating from the same premise. This way annotators must only read the premise once, to compare it with those newly created sentences. Not all our identiffied categories of word-pairs are well represented. In order to get the most of our less frequent categories, we sample the 10,000 sentence-pairs, that need to be annotated, in a greedy manner: After soring categories by the amount of samples, that we created, we start with the least representative categories and sample as many sentence-pairs as possible, keeping our constraint of needing a multiple of five hypothesis per premise up to an upper bound of samples per category[54]. If more samples are required to complete a HIT with five hypothesis, the next categories are checked in the order of representativeness. Furthermore we keep track of the amount of each word-pair, that is included in the sampled sentence-pairs, and always prefer less-frequent word-pairs, if several options exist.

Annotation process

To simplify the HITs for annotators, such that they do not need to understand the labelling scheme of SNLI, we create a set of questions, highly aligned[55] with those, proposed of Bowman et al. (2015), that we later map to entailment labels. Specifically, we ask:

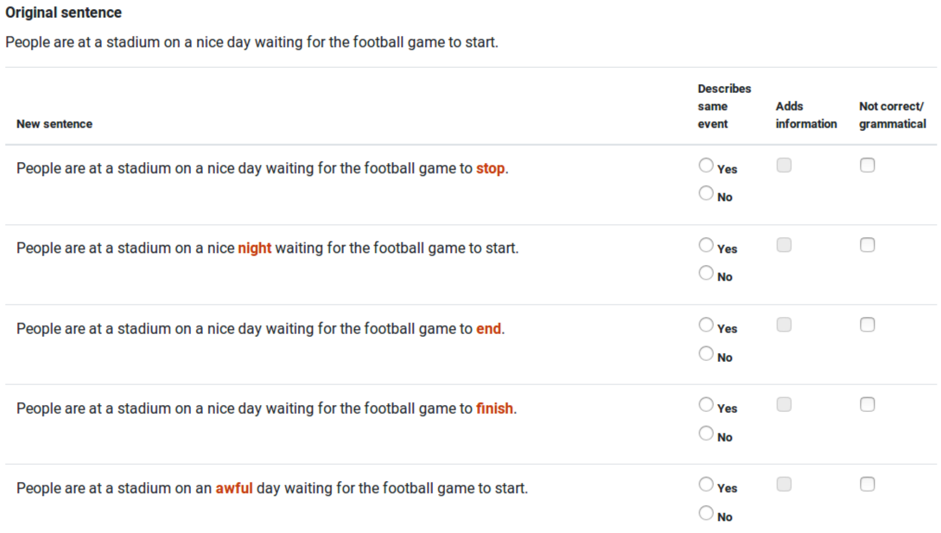
1. if both sentences describe **the same event**
2. if the hypothesis **adds new information**
3. if the sentence is **invalid**

[53] https://www.mturk.com/

[54] The upper bound is always re-calculated, serving the purpose of not sampling too much of one category, since the amount of all samples is bound at 10,000.

[55] Bowman et al. (2015) only provide their annotation guidelines for the task of creating new hypothesis, not for validating them. The validation task was conducted separately and was only open for annotators who participated in the hypothesis-generation task and thus were qualified already.

Samples that are answered negatively for (1) result in the label *contradiction*. If (1) is answered positively, the label is either *neutral*, if (2) is answered positively, or *entailment*, if (2) is answered negatively. Sentences that show major grammatical errors or do not make sense to a native English speaker, should be marked using (3) and no label is inferred. We defined the HIT user interface such that no other combinations can be selected as an answer. The questions are explained in deeper detail in an additional introdcution, and we ensure they are understood correctly with a mandatory qualification test. Since SNLI does contain grammatical and spelling errors, we specifically allowed minor errors of that kind. Additionally, SNLI does contain fictive sentences that are not realistic. Yet, it is hard to define the degree that sentences are allowed to sound unrealistic. While "flying to school" instead of "walking to school" may a bit unrealistic but still plausible for a fictive scenario, "sitting at the table and eating" and "walking at the table and eating" seems indeed very unlikely. This however, is highly dependant on the subjective perspective. We defined this aspect of (3) rather swammy, by allowing fictive scenarios, however counting on the English capabilities of native speakers, to identify if those sentences make sense to them and seem like a proper usage of words. Due to this loose definition, we strictly ignore all samples, if at least one single annotator marked them as invalid, not requireing a majority in this case. To assure, that our workers have shown to annotate appropriately to the task description, we only accept annotators with an approval rate of at least 99% and a minimum of 1,000 prior tasks. A sample HIT is shown in Figure 23. The questions are explained to the users with example HITs of the same form in the instructions. In order to make the task more attractive to annotators, without increasing the payment too much, we simplified the process by highlighting the exchanged words.



Original sentence

People are at a stadium on a nice day waiting for the football game to start.

New sentence	Describes same event	Adds information	Not correct/ grammatical
People are at a stadium on a nice day waiting for the football game to **stop**.	○ Yes ○ No	□	□
People are at a stadium on a nice **night** waiting for the football game to start.	○ Yes ○ No	□	□
People are at a stadium on a nice day waiting for the football game to **end**.	○ Yes ○ No	□	□
People are at a stadium on a nice day waiting for the football game to **finish**.	○ Yes ○ No	□	□
People are at a stadium on an **awful** day waiting for the football game to start.	○ Yes ○ No	□	□

Figure 23: Example of a HIT in Amazon Mechanical Turk.

We assign each HIT to three annotators. After removing all invalid sentence-pairs, we consider the majority label as the gold label, if at least two annotators agree on it. Sentence-pairs without agreement are not considered for our new testset. After an inital annotation round of 1,000 samples, we remove categories and word-pairs, that show a high tendency of having invalid sentences. We show the statistics of our final testset, consisting of 8,193 sentence-pairs in Table 14. As can be seen, the new test set is heavily focused on contradicting samples. Since this test set only serves, to measure the capabilities of a model in dealing with lexical inferences (and NLU with reduced dataset-specific patterns), but not to replace the original SNLI testset, this is not problematic. It still must be considered, when evaluating a model, as a simple baseline, predicting everything as contradiction, would result in an accuracy of 87.4%. We report the agreement with Fleiss Kappa (Landis and Koch, 1977) (over all samples and for the representative labels *entailment* and *contradiction*), as it was done by Bowman et al. (2015) for the original SNLI. For better comparison, we re-calculate the numbers on all valid samples of the SNLI testset. The new testset yields "substantial agreement" with a Fleiss Kappa of 0.61. It should be noted, that Kappa also considers, how likely a specific label would selected by chance, and for this purpose takes

	Instances				**Fleiss κ**		
	contradiction	neutral	entailment	**Overall**	contradiction	entailment	**Overall**
SNLI Test	3,236	3,215	3,364	**9,815**	0.77	0.69	**0.67**
New Test	7,164	47	982	**8,193**	0.61	0.90	**0.61**

Table 14: Statistics of SNLI testset compared with the newly generated testset.

the overall label distribution into account. This does not influence the measure for the original SNLI, since all labels are evenly distributed. For the new test set however, this method assumes that contradiction is more likely to appear by chance, due to its high frequency. As this results from our selected word-pairs rather than an underlying "natural distribution", the figure, calculated by Fleiss Kappa, might be less suited as an agreement measure for the new dataset. Yet, for very different reasons, it is indeed likely that annotators are slightly biased towards selecting contradiction, as the result of our HIT presentation: Since most word-pairs are contradicting and word differences are highlighted, annotators might, to some extend, shift their focus more on the difference between those words, rather than solely on the actual word-usage in context. To provide an easier to interpret measure of this dataset, we estimate the human performance in the same manner, as done by Gong et al. (2017) for SNLI. They consider all samples with majoriy label, which results in the gold label, and calculate the ratio of annotator labels, matching the majority gold label, as the estimate for the human performance in terms of accuracy. Thus, let $g(x,y)$, with x as the annotator label and y as the estimated gold label, define, if the annotation counts as a misclassification or not:

$$g(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \tag{9}$$

Let furthermore L contain all pairs (x,y) of the annotated dataset and $|L|$ be the amount of elements within L. Following Gong et al. (2017), we estimate the human performance a in accuracy as follows:

$$a = \frac{\sum_{(x,y)\in L} g(x,y)}{|L|} \tag{10}$$

Doing so, we estimate the human performance on the new testset to be 94.1%, slightly higher than the human performance estimated on SNLI with only 87.7%, indicating that our new sentence-pairs do not pose additional difficulties, but in fact indeed seem relatively easy for humans.

5.3 Evaluation

We evaluate three neural models without external knowledge other than the one from distributed word-embeddings, that achieve strong results on SNLI. All models are retrained with different datasets, using the provided code and keeping all hyperparameters. We explain the experimental setup and results below.

5.3.1 Experimental setup

Models without external knowledge

We evaluate the Residual-Stacked Encoder$^\diamondsuit$ (Nie and Bansal, 2017), as explained in Section §2.3, ESIM (Chen et al., 2017b) and Decomposable Attention (Parikh et al., 2016), both explained in Section §3.3. The published model of ESIM ensembles two models with different sentence encoding strategies, one is based on a TreeLSTM, the other on a biLSTM. For our experiments we retrain only the biLSTM-based model. For Decomposable-Attention, we use the AllenNLP re-implementation[56]. As opposed to the reported version on the SNLI leaderboard[57] this implementation does not use the optional intra-sentence attention. Its performance on the SNLI test is with 84.7% slightly lower, but comparable to the model with intra-sentence attention (86.3%). All models have different characteristics, depticted in Table 15 and are at the time of the experiment amongst the best within their categories.

	Finetune Embeddings	**LSTM-based**	**Inter-sentence Attention**
Decomposable Attention (Parikh et al., 2016)	–	–	*yes*
Residual-Stacked Encoder Nie and Bansal (2017)	*yes*	*yes*	–
ESIM (Chen et al., 2017b)	*yes*	*yes*	*yes*

Table 15: Architectural comparison of tested neural models without external knowledge.

[56] http://allennlp.org/models

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

5.3.2 Models with external knowledge

Additionally we also provide a simple WordNet baseline, predicting the relationship of (p,h) by assuming all sentences to be upward-monotone and, thus having the same relation as (w_p,w_h). Specifically, for each (w_p,w_h) we check their lexical semantic relation within WordNet, and map it to a relation label in the following manner:

- **Synonymy:** Synonyms are predicted as *entailment*.
- **Antonomy:** Antonyms are predicted as *contradiction*.
- **Hypernomy:** If w_p is a hypernym of w_h the sentence-pair is predicted as *neutral*, if w_p is a hyponym of w_h as *entailment*.
- **Co-hyponomy:** Cohyponyms are predicted as *contradiction*. We only only consider co-hyponyms with a maximum distance of two edges in the ontology to their common hypernym, as considering all potential co-hyponyms would yield all (w_p,w_h) coming from the same (very general) root-word to be labelled as contradiction.

We map multi-word expressions like “at night” to their meaning-carrying word (“night”), if the function words have only been added for a higher precision, when replacing the words in the generation process. In case they refer to actual entities (e.g. “New Zealand”), we identify the applicable synsets of the whole expression in WordNet. As explained in Section §3.1.1, words may have several synsets leading to potentially several different lexical relations amongst the words of interest w_p and w_h. For each relation between both words, we calculate a score $s = \max(r_p, r_h)$ with r_p and r_h being the rank of the synset of the word w_p and w_h respectively. This follows the common heuristic that dominant senses appear as the first synsets while rare senses appear at the end (McCarthy et al., 2004). Subsequently, if several lexical relations exist, we consider the one with the lowest assigned score as tie-breaker[58] and thus, tend to focus on more dominant word-senses. Of course, this baseline only is possible, if knowing that p and h only differ in w_p and w_h and is thus not applicable to sentence-pairs in general. Yet it provides insight, to what extend information within WordNet can help on our new test set. In addition to that we, also report[59] the results of KIM (Chen et al., 2017a), as explained in Section §3.3.2.

Traing data

In addtion to training all models on SNLI, wich is considered relatively easy, we also train each model on the union of SNLI and MultiNLI and SciTail respectively, both are assumed to be more difficult and explained in deeper detail in Section §3.2. The motivation is, that while SNLI might lack the training data needed, to learn the required lexical knowledge, this data may be available in the other datasets, which are presumably less simple.

5.3.3 Results

The results for each model with the according train sets are visualized in Table 16. There is a clear trend, that adding MultiNLI to the training data boosts the model’s performance on the new test set. At the same time it decreases the test accuracy on SNLI, indicating that the performance, gained on SNLI test, does not reflect the true NLU capabilities, since clearly less lexical semantic relations are understood. Yet, compared with the original estimated performance, even by almost doubling the amount of train data with MultiNLI, all models without external knowledge show a significant drop in performance. While MultiNLI follows the same labelling scheme as SNLI, and thus is compatible, SciTail does not specifically assume event coeference and lacks having the label *contradiction*, which is dominant in the new test set. Hence, the models seem to not be able to leverage from the extended amount of data in this case. Both models with WordNet information perform significantly better than the ones without. This shows, that the lexical relations, as contained in WordNet, are sufficient to gain strong improvements on the new dataset. In addition to that, those relations have also shown to be useful for KIM in training on SNLI (as they are considered for the prediction). This especially shows some crucial drawbacks in the neural models without WordNet, as they clearly lack to extract those features when trained solely with textual input, by learning features based on arbitrary patterns at the cost of also meaningful (for SNLI and naturally for NLU) features of lexical relations. Only dropping by 5.1 points in accuracy w.r.t. the performance on SNLI, KIM seems substentially more stable, when sentences are adapted, forcing the model to predict based on a language understanding rather than dataset specific patterns. This of course may be due to the fact, that the new testset, being closely related to the train data and requiring the same knowledge, as added for KIM, is highly suitable forKIM. Thus, it still does not prove truely superior NLU, yet it shows that they found a successful strategy of integrating this resource into neural models.

[58] In case the score s is identical for several relations, we select the relation in the following order ($X > Y$ meaning X is preferred over Y): synonym > antonym > hypernym > hyponym > co-hyponym

[59] We did not conduct the experiment with KIM ourselves, but received their results from the orignal authors recently. The analysis part thus does not contain deeper analysis of the performance of KIM.

Model	Train set	SNLI test set	New test set	Δ
Decomposable Attention (Parikh et al., 2016)	SNLI	84.7%	51.9%	-32.8
	MultiNLI + SNLI	84.9%	65.8%	-19.1
	SciTail + SNLI	85.0%	49.0%	-36.0
ESIM (Chen et al., 2017b)	SNLI	87.9%	65.6%	-22.3
	MultiNLI + SNLI	86.3%	74.9%	-11.4
	SciTail + SNLI	88.3%	67.7%	-20.6
Residual-Stacked Encoder$^\diamond$ (Nie and Bansal, 2017)	SNLI	86.0%	62.2%	-23.8
	MultiNLI + SNLI	84.6%	68.2%	-16.8
	SciTail + SNLI	85.0%	60.1%	-24.9
WordNet Baseline	-	-	85.8%	-
KIM (Chen et al., 2017a)	SNLI	88.6%	83.5%	-5.1

Table 16: Results of models on the new test set compared with the original SNLI test set.

5.4 Analysis

We take a closer look on the performance achieved by the models without external knowledge. As the performance only improves marginally, if the train data is tremendously increased by adding MultiNLI, we focus our analysis part on the models, solely trained on the original SNLI dataset.

5.4.1 Accuracy by category

Table 17 shows the accuracy per category, as defined when creating the word-pairs, for all models, including KIM and the WordNet baseline. As not all categories contain an even amont of sanples, we additionally supply information about this figure, together with sample words for a better understanding, what each category represents. Originating from our word-pairs, almost all categories are majority labelled contradiction, solely synonyms mostly are labelled as entailment. All neural models achieve good results on categories that occur very frequently within SNLI in general, like *colors*. Also *instruments* are well captures. We find that muscic instruments often occur in SNLI in very similar sentences containing some kind of `actor` and in conjunction with the instrument and the verbs "hold" or especially "play". In contradicting samples of the train data, in most cases the instrument changes. Thus our newly created sentence-pairs are very similar to those within the train data, explaining the good performance within that category. On the other hand, categories that are rare in SNLI, like *planets* or *ordinals*, are not well understood by those models. As opposed to to *instruments*,

Category	Amount	Example Words	Decomposable Attention	ESIM	Residual Encoders	WordNet Baseline	KIM
antonyms	1,147	*loves - dislikes*	41.6%	70.4%	58.2%	95.5%	86.5%
cardinals	759	*five - seven*	53.5%	75.5%	53.1%	98.6%	93.4%
nationalities	755	*Greek - Italian*	37.5%	35.9%	70.9%	78.5%	73.5%
drinks	731	*lemonade - beer*	52.9%	63.7%	52.0%	94.8%	96.6%
antonyms (WN)	706	*sitting - standing*	55.1%	74.6%	67.9%	94.5%	78.8%
colors	699	*red - blue*	85.0%	96.1%	87.0%	98.7%	98.3%
ordinals	663	*fifth - 16th*	2.1%	21.0%	5.4%	40.7%	56.6%
countries	613	*Mexico - Peru*	15.2%	25.4%	66.2%	100.0%	70.8%
rooms	595	*kitchen - bathroom*	59.2%	69.4%	63.4%	89.9%	77.6%
materials	397	*stone - glass*	65.2%	89.7%	79.9%	75.3%	98.7%
vegetables	109	*tomato -potato*	43.1%	31.2%	37.6%	86.2%	79.8%
instruments	65	*harmonica - harp*	96.9%	90.8%	96.9%	67.7%	96.9%
planets	60	*Mars - Venus*	31.7%	3.3%	21.7%	100.0%	5.0%
synonyms	894	*happy - joyful*	97.5%	99.7%	86.1%	70.5%	92.1%
total	8,193		51.9%	65.6%	62.2%	85.8%	83.5%

Table 17: Accuracy reached for the tested models for each category with assoziated sample words and the amount of instances.

the relevance of *ordinals* within a sentence in SNLI usually is less crucial. Yet this originates only the commonly applied strategies by the annotators, as identified by Gururangan et al. (2018). Consider for instance the sentence “"A man racing his motorcycle comes in *first*."”, which naturally yields in contradiction if we replace “first” by “eights”. Yet the model without external information seemingly do not differentitate between both ordinal numbers, as annotators rather tend to change “man” or even “motorcycle”, but not “first”. Also *drinks*, *vegetables* and *rooms* are generally harder for the model to predict for similar reasons. The reason for *antonyms* being more difficult than *antonyms_WordNet* most like arises from the fact, that they include gender-related antonyms, appearing in SNLI in abundance, that we do not include within the hand-crafted word pairs. As *synonym* examples have large lexical overlap and differ only by one word, occuring in similar contexts (and subsequentially having a similar word-vector), it is no surprise, that all models achieve a good performance here. Inter-sentence-attention seems to have an advantage in this case, since only the Residual-Stacked Encoders does not improve significantly over its original test performance. We thus focus the remaining part of the analysis on contradicting sentence-pairs only.

5.4.2 Impact on the word embeddings

As previously pointed out, word-pairs with the lexical semantic relations, antonymy and co-hyponynomy, in many cases result in similar word representations with distributed embeddings. Subsequentially, we first analyse the impact of those word-representations, leveraging from the fact, that sentence-pairs in our new dataset only differs in one known word.

Without fine-tuned embeddings

Figure 24 visualizes the performance of all contradicing samples, compared to the cosine similarity between the word-vectors of w_p and w_h, achieved by Decomposable Attention. We exclude any multi-word expressions in this analysis.

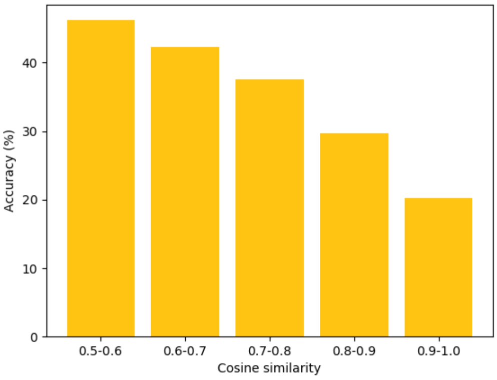


Figure 24: Accuracy by cosine similarity reached by Decomposable Attention (without fine-tuned embeddings).

Let v_p and v_h be the word vectors of the GloVe embeddings, used by Decomposable Attention, the according cosine simmilarity $\cos(v_p, v_h)$ is calculated as follows:

$$\cos(v_p, v_h) = \frac{v_p v_h}{|v_p||v_h|} \quad (11)$$

We observe, that without fine-tuned embeddings, the reached accuracy highly correlates with the similarity of the word representations even though Decomposable Attention uses only the lower-cased word embeddings and thus contains comparably more samples per word-vector than the other two models, ESIM and Residual-Stacked Encoder. Those models rely on cased word-embeddings, and we could not find the same correlation between the accuracy and word-similarity. We assume this stems from the fact, that both models fine-tune embeddings and thus push contradicting words, as seen in the training, further apart in the embedding space.

With fine-tuned embeddings

We evaluate the accuracy of both models with fine-tuned embeddings w.r.t. the amount w_p and w_h seen during training on SNLI and visualize the results in Figure 25. The numbers on the x-axis show the upper bound of word occurences of the word-pair, denoted as $a_{(w_p,w_h)}$, responsible for creating each sentence-pair. Specifically, we calculate the $a_{(w_p,w_h)}$ by taking the more frequent word within SNLI train data, thus $a_{(w_p,w_h)} = \max(a_{w_p}, a_{w_h})$, with a_{w_p} and a_{w_h} being the amount of sentences, containing w_p and w_h respectively. In case, one of w_p or w_h is an expression containing

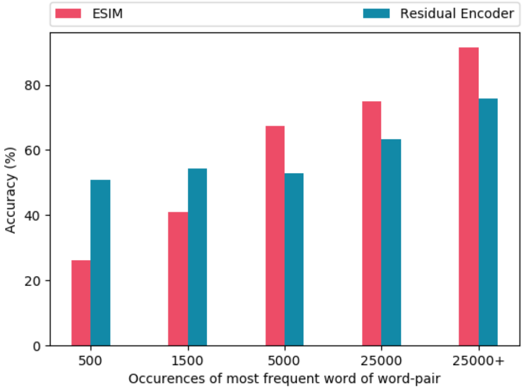


Figure 25: Accuracy by word freqency for Residual-Stacked Encoder and ESIM.

multiple words, denoted as $e = [w_1, \ldots, w_{n-1}, w_n]$, we calculate the according amount, denoted as a_e, by considering the least frequent word: $a_e = \min(w_1, \ldots, w_{n-1}, w_n)$. Intuitively, this will ignore added function words and, in most cases, focus on the meaning-carrying word within the expression. Both models depend on the same underlying GloVe word-embeddings, yet it seems that Residual-Stacked Encoder$^{\Diamond}$ achieves better results on less frequent words. While still performing considerably worse with fewer examples seen, the more individually created sentence representations, as trained in this model, generalize better for sparcely present lexical relations. As opposed to the Residual-Stacked Encoder, ESIM seems to heavily depends on a high frequency of the words to classify sentence-pairs correctly. While ESIM performs very poor for less frequent words, it shows to quickly increase in performance with an increasing amount of samples, containing the same words, within the train data. This presumeably arises from the inter-sentence attention, aligning words from both sentences with each other. The resulting sentence representations are therefore less general and suited for the individual word relations of both sentences, leading to a higher performance, if similar word relations have previously been seen in training; at the same time however, reducing the generalization capabilites. Subsequentially we take a closer look into the performance of ESIM, the best of all three evaluated neural models without external knowledge, and compare the amount of similar samples seen during training with the reached accuracy. We count samples (p, h) from the train data with the gold label contradiction and consider them, if they contain w_p in p and w_h in h, *similar* to all samples in the new dataset, arising from (w_p,w_h). The results are shown in Table 18. It can be seen, that

Frequency	0	1 – 4	5 – 9	10 – 49	50 – 99	100+
Accuracy	40.2%	70.6%	91.4%	92.1%	97.5%	98.5%

Table 18: Accuracy by the amount of similar samples in SNLI train data for ESIM on contradicting samples.

indeed, the performance of ESIM is high, if it has seen w_p and w_h in a contradicting context in a sufficiently high amount, whereas it performs poorly, if it has not seen both words within a contradiction-labelled sample at least once. This shows, that the comparably higher performance of ESIM is matter of memorizing (w_p,w_h) as being contradictive, rather than generalization over similar constructs. Similarily, this explains the increase in accuracy when adding MultiNLI, not because it learns to generalize better, but because it has seen slightly more contradictive word-pairs in a contradicting context. While this is sufficient to achieve a high performance on SNLI, we show that the main goal of machine learning, to generalize over unseen textual constructions, in this case is not met.

5.5 Conclusion of the adversarial dataset

We show that state-of-the-art NLI systems with only pretrained word embeddings as external information trained on any of these datasets are limited in their generalization ability and fail to capture simple inferences. Additionally, we show, that the SNLI test set alone is not a sufficient measure of the languge understanding capabilities of a model, using our newly created test set. As the high performance arises from arbitrary patterns, that we excluded in our dataset, this number is in fact somewhat misleading, considering that NLI originally is meant to be closely related to NLU capabilites of the models (Williams et al., 2017). While models without external knowledge perform good by memorizing patterns, the relevant information for the new testset are also useful for SNLI and improve the model's generalization abilities. Both knowledge-rich approaches achieve comparably high performances on the new dataset, but still have a potential for improvement. Those may either be tackled by a resource with a higher coverage than WordNet, or by improving lexical inference in context (Shwartz and Dagan, 2016).

6 Approaches to incorporate WordNet information

Having the dataset of the previous section, we next try to improve our latest re-implementation Residual-Stacked Encoder† using WordNet.

6.1 Methods

Unlike KIM, which has shown an intuitive and successful strategy of incorporating WordNet into a neural network, the Residual-Stacked Encoder does not use inter-sentence attention. Without changing this, we therefore cannot likewise align words of p with words of h to identify their WordNet relation. We intend to leave the model with the plain sentence-encoding architecture, targeting the incorporation of external resources for general sentence representations thus by encoding each sentence individually (Nangia et al., 2017). Naturally, not being able to align words with each other poses a new difficulty, since the relations of WordNet are defined between two senses. Subsequentially, we apply other strategies, than directly encoding the relation of two words (or senses), explained below.

6.1.1 Drawbacks of using insights of max-pooled sentence representations

In Section §4 we gained valid insights on the sentence representations and showed that these sucessfully can be used to change the meaning of sentence representations. Following these conclusions, a possible strategy is, to train the model in a way, that antonyms or co-hyponyms result in distinct high dimensions within the sentence representation, synonyms having the same high values and hypernyms a subset of hyponyms. Knowing reasonable values for each values within the dimensions, this could be broken down to a simple regression problem. Since we did not find an elegant way to naturally include this into the loss function, two possibilities exist:

- **Manually define the ξ of dimensions:** The values for each word are defined before the training, w.r.t. the lexical relations that are shared amongst them, using automatic optimization techniques. The loss is defined w.r.t. those defined values for each word within a sentence.

- **Regularily analyse the ξ of dimensions:** Instead of defining the ξ of each dimension beforehand, it can be gained by an anaylse step, similar to the one conducted for this work, every iteration. Knowing the dimensions, how they are already used by the model, they can slightly be adapted such that our wanted criteria is met. For instance, let the words w_1 and w_2 share a common close hypernym, thus both are co-hyponyms. After identifying the dimensions, that result in a high value coming from both words, some of tose dimensions are reserved for w_1 only, while others are reserved for w_2 only. This may also be achieved automatically, if dimensions are analysed for their high-valued words w.r.t. WordNet relations.

Both approaches severely lack the possibility of encoding words depending on their context into the sentence representation. While we gained some insights on the sentence-representations, it is insufficient to define them in a reasonable way. Even if this would be possible, it is not desired. Both strategies, especially the first, highly resemble traditional feature-engineering. Since the automatic feature selection is one of the key strengths of neural models (Bengio et al., 2013), those strategies would rather be similar to a step backward than forward. Instead we identify to potential strategies, that are simpler to implement and would result in a broader applicability, not being tied to max-pooled sentence representations.

6.1.2 Fuse WordNet information within the embedding-layer

Additional information within the word-representations has the high advantage, of being very gerneral applicable. Following Rücklé et al. (2018) we do not use exclusively retrained or adjusted word-embeddings. Instead, for each word w we lookup the according word-vector within the original distributed GloVe embeddings and concatenate it with the corresponding word-vector of the same w from the additional word-embeddings. If no corresponding vector for w is present within those, we concatenate an zero-valued vector of the same dimensionality. Doing so, we do not chnge the original information of distributed word-embeddings such that the model may still rely on the same features, if they are useful. Even though some of the newly introduced might be redundant w.r.t. the orignal GloVe embeddings, some contain additional information, that the network can use to differentiate between words, that are highly similar in GloVe but mutually exclusive. Additional to doing this experiment with the mono-lingual attract-repel vectors, provided by Rücklé et al. (2018), we use two different word-vecor sources.

Overfitting WordNet

We apply a simple method to create addtitional word vectors v that are similar for the words w_1 and w_2, if they are synonyms within WordNet, and different if w_1 and w_2 are antonyms or co-hyponyms. For this we extract samples (w_1,

w_2, relation), whereas w_1 and w_2 are lemmas, that are linked via relation within WordNet, represented by their original GloVe embeddings. Using a two layer MLP, we map each word-vector $w \in \mathbb{R}^{300}$ to $v \in \mathbb{R}^{20}$. In our last layer we apply tangens-hyperbolicus as non-linearity, to squeeze all values v^i within v, with v^i being the ith value within v, are in an appropriate range: $\forall i : [i \in \{x \in \mathbb{N} | x < 20\} \Rightarrow v^i \in \{x \in \mathbb{R} | -1 < x < 1\}]$. Let $w \in \mathbb{R}^{300 \times 1}$ be the original word-embedding, $W_1 \in \mathbb{R}^{100 \times 300}$ and $b_1 \in \mathbb{R}^{100 \times 1}$ the weight matrix and bias of the first layer, and $W_2 \in \mathbb{R}^{20 \times 100}$ and $b_2 \in \mathbb{R}^{20 \times 1}$ of the second layer respectively. The new word-vector v is calculated as:

$$v = \tanh(W_2 \text{ reLU}(W_1 w + b_1) + b_2) \tag{12}$$

We optimize the representations v_1 and v_2, coming from (w_1, w_2, relation) using Mean Squared Error (MSE) with the Eucledian Distance, which should be high, if the relation indicates, w_1 and w_2 are mutually exclusive, and low, if both are synonyms. Thus, we define $\theta = 0$ for synonyms, and $\theta = 10$, for antonyms and co-hyponyms rspectively (we bound the difference to $\frac{|v|}{2}$, with $|v|$ being the amount of dimensions of the new word-vectors, as it creates sufficiently distinct vectors):

$$\text{loss} = \frac{1}{2}\left(\sqrt{\sum_{i=1}^{20}(v_1^i - v_2^i)^2} - \theta\right)^2 \tag{13}$$

We overfit on the lexical relations, extracted from WordNet, using the loss function in Equation (13), intending to memorize whether two words are compatible or not. This optimization process also updates the GloVe embeddings of w during training. In order to create word-vectors that specifically focus on either antonymy or co-hyponomy, we train embeddings for each of those relations individually, both times together with synonyms in order to have a counterpart. Embeddings differentiating between synonyms and antonyms are referred to as `Trained-Syn-Ant`, differentiation between synonyms and co-hyponyms as `Trained-Syn-Cohyp`. We refer to the concatenation of both embeddings as `Trained-Syn-Cohyp-Ant`. Since the Eucledian Distance is a symmetric measure we cannot include the hypernym-hyponym relation into those embeddings, due to their asymmetry. We thus train other embeddings differentiation between all relevant lexical semantic relations[60] in the same manner, but with 50 instead of 20 dimensions. In order to enable the network to deal with asymmetric relations (and thus is able to predict that a w_1 is a hypernym of w_2, whereas w_2 is a hyponym of w_1) we apply a softmax layer and train the network with cross-entropy loss, prediction the actual relation, holding between w_1 as the first input and w_2 as the second input. We refer to those embeddings as `Trained-All`.

Adding categorical information

Alternatively, especially targeting the detection of co-hyponyms, instead of concatenating different embeddings, we conctatenate each word w with the distributed GloVe vector of the hypernyms of each w. The motivation is, that the network is able to identify, that two words share the same hypernym and hence, learning more subtle differences within the actual value-differences within both word-vectors. Due to WordNet's fine-grained ontology, we apply different strategies of taking the mean vector of one up to five hypernomys of a word. If hypernyms within WordNet consist of multi-word expressions, we take the word-embedding of the last word. For instance for "single-reed instrument", being the first hypernym of "saxophone", we concatenate the word-vector of "saxophone" with the word vector of "instrument". We refer to those embeddings as `Hypernyms-<amount-of-hypernyms>`.

6.1.3 Fuse WordNet information within the sentence-representations

It is very known, that neural networks do well in learning relevant features (Bengio et al., 2013), however, as seen in Section §5 and shown by Gururangan et al. (2018), those features do not necessarily correspond with NLU, but are heavily biased by dataset-specific patterns. These are not reduced, if we add additional informtion to the embeddings, thus we still need to rely on the full model to pick up on good features, yielding to correct decisions for the *right* reason. Gülçehre and Bengio (2016) show on a very different task, of detecting pentomino shapes, that deep neural networks may not even find the most useful features and can heavily leverage from human guidance when creating intermediate representations. In their very simple toy-scenario, this could be done by manually creating intermediate target representations, which is not easily possible for sentence-representations in NLP. We thus continue training the nural network in an end-to-end manner. In order to guide the network in learning more useful sentence-representations, we create a second task, namely the helper-task, sharing some basic components with the maintask (predict the label for NLI). Thus, both tasks rely on the same sentence representation, that will therefore encode relevant features for both tasks, whereas one can leverage from the features from the other. This is commonly known as multitask-learning and has shown to be successful to improve the generalization of shared representations (Nangia et al., 2017).

[60] Synonymy, Antonomy, Hypernyomy, Co-hyponomy

Multitask architecture

The multitask setup is visualized in Figure 26. The left side shows the standard architecture of the Residual-Stacked Encoder, as defined in Section §2.3 Both sentences p and h are encoded using the same sentence encoder. The resulting sentence representations are concatenated with the additional features and classified by the final MLP into on the three labels entailment, neutral or contradiction.The additonal MLP on the right side is used for the helper task. We create the helper-task with the intention to force the model to encode differences between two words w_1 and w_2, if one is the antonym or co-hyponym of the other, into the sentence representations, as this essentially is used for the final prediction. Likewise, if w_1 and w_2 are synonyms or w_2 is the hypernym of w_1 and thus entailed by it, we want this information to be encoded as well. In order to contain this information, which preumeably helps the generalization power of the main-task, within the sentence representation, we define our helper task, as a binary classification problem, whether a word is encoded within the sentence representation or not. For this, we consider both sentences p and h from a BoW perspective. Let $S = \{w^0, w^1, \ldots, w^{n-1}, w^n\}$ be the set of all n distinct words w within a sentence. We apply the same task for p and h. Since we do not consider them simultaneously but individually, we define the helper-task using the general S respectively for both. For each $w^i \in S$, we identify words from WordNet, being related to w^i with one of the previously mentioned lexical semantic relations. Let A be the set of words, who's meaning must be entailed by the sentence representation, thus A contains all hypernyms and synonyms of all $w^i \in S$. Similarily, let B be the set of words, who's meaning is *not* entailed by the sentence, thus antonyms and co-hyponyms of all $w^i \in S$. Additionally, all w^i that have related words via lexical semantic relations are also added to A. A sentence like "People are watching a soccer game between Brazil and Mexico." may still cause conflicts, as A contains "Brazil" and "Mexico", however they may also be present within B, being mutual co-hyponyms. To avoid conflicts, if several co-hyponyms are present within the same sentence, we ensure that A and B are not overlapping, by setting $B = B \setminus (S \cup A)$. The final helper-task takes a sentence representation r and a word embedding e as input, and must classify, whether e belongs to A or B, meaning whether e is present (in the sense of entailment) within r or not. Since the same embeddings are used and fine-tuned, this may also be seen similarly to further postprocess word vectors, like in Attract-Repel (Vulić and Mrkšić, 2017), but additionally ensuring, those differences are propagated into the sentence representation.

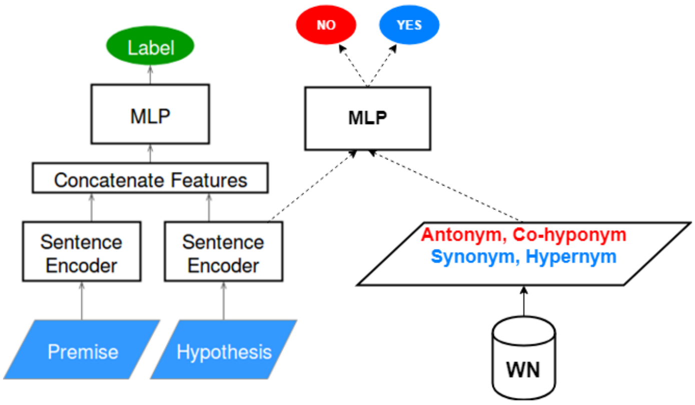


Figure 26: Architecture of the Residual-Stacked Encoder with multitask learning for the sentence-representations.

Training

The main-task and the helper-task are simultaneously trained. Thus, the combined loss, denoted as $loss_{\text{combined}}$, aggregating the loss for the main-task, denoted as $loss_{\text{main}}$, and for the helper task, denoted as $loss_{\text{helper}}$, as

$$\text{loss}_{\text{combined}} = \alpha \ \text{loss}_{\text{main}} + (1 - \alpha) \ \text{loss}_{\text{helper}} \tag{14}$$

whereas $\alpha \in \{x \in \mathbb{R} | 0 \leq x \leq 1\}$ regulates the impact of the main-task, with a high value ($\alpha = 1$) only considering the main-task and a low value ($\alpha = 0$) only the helper-task. We train the network using a batchsize of 32. While $loss_{\text{main}}$ remains the original mean cross-entropy, $loss_{\text{helper}}$ is also based on mean cross-entropy, yet is down-weighted for the following reason. Let A_p, B_p and A_h, B_h be A and B according to the definitions above for p and h respectively and $|A|$ denote the amount of samples within a set A. One can safely assume, that the amount of samples for the helper-task is tremendously higher than for the main task, as one single sample (p,h) here yields $|A_p| + |B_p| + |A_h| + |B_h| \gg 1$ samples in the helper-task. Let b be the batch-size and p^i and h^i denote the ith p or h within a minibatch. We calculate n to be the total amount of samples for the helper-task within a given batch:

$$n = \sum_{i=1}^{b} \Big(|A_{p^i}| + |B_{p^i}| + |A_{h^i}| + |B_{h^i}| \Big) \tag{15}$$

Let $loss_s$ be the loss, calculated with mean cross-entropy, for all samples coming from A_s, B_s, with s being any sentence p or h. The re-weigted $loss_{\text{helper}}$ over all (p, h) within a minibatch is calculated as:

$$\text{loss}_{\text{helper}} = \sum_{i=1}^{b} \Bigg(\frac{|A_{p^i}| + |B_{p^i}|}{n} \text{loss}_{\text{p}^i} + \frac{|A_{h^i}| + |B_{h^i}|}{n} \text{loss}_{\text{h}^i} \Bigg) \tag{16}$$

To reduce training time, we always train the samples for the helper-task and main-task, based on the same source sentences, within the same batch (as opposed to iterate between both tasks), keeping the frequency of the cost-intensive sentence-encoding step to a minimum.

Multitask variations

We evaluate several implementations with small differences or changed hyperparameters, following the presented architecture. Those are described below, mostly differing in their impact on the sentence-representation.

- **Size and amount of layers:** We evaluate different sizes of the helper-task MLP. Naturally, the simpler the helper network (fewer layers and dimensions per layer), the more relevant information must be encoded within the representation. This should be preferrable, since finally we aim for the main-task to identify the same, hopefully meaningful, features, and we do not have further usage for the logic, implemented within the helper-task MLP.

- **Dropout:** Similarily, by using dropout in the helper-task, we motivate the creation of redundant features within the sentence-representation, by applying dropout to the helper-task (0.1).

- **Re-sample less frequent label:** We observe that usually $|A_s| < |B_s|$, resulting especially from the large amount of co-hyponyms. In order to prevent the helper task to take the label distribution into account rather than creating relevant features, we re-sample the less frequent class of A_s or B_s, such that $|A_s| = |B_s|$.

- **Reweighting tasks:** By either statically adapting the value of α beforehand, or dynamically updating it during training, we change the impact of each task. Specifically, we enable in the `finetune`[61] setting, to only optimize the performance on SNLI after initially focusing on both tasks equally. In the setting `focus-start`[62], the encoder focuses on the helper task in within the first iterations. Only after the sentence representation is useful for the helper task, the main task is also considered. As opposed to that, in `focus-mid`[63], we first tune the sentence-representation only for SNLI, adapt this useful representation then for the helper task and finally finetune to SNLI again.

- **Freeze helper-task weights:** To not encode too much logic in the helper MLP, we freeze the weights after one iteration, dentoted as `(freeze)`, assuming this is sufficient to have a reasonable weight matrix learned. Subsequent enhancements must afterwards be encoded directly in the sentence representation.

- **Additional weight matrix on top of sentence-encoder:** In order to achieve a strong accuracy for the helper task, more than one layer is required in the MLP. To reduce the impact of the unused layers, we use a two layer MLP for the helper task, the main task however does not dpeend on the original sentence-representation anymore, but from the output of the first layer of the helper-task MLP. We name this configuration `(shared)` in the evaluation.

- **Focus on responsible words:** Instead of using all values, encoded within the sentence-representation, we follow the same approach as described in Section §4, identifying which word is responsible for which dimension, and focus the network on those explicit dimensions. Thus, we consider the original pairs (w_1, w_2, `relation`) and set all dimensions within the sentence-representation to zero, if they do not arise from w_1. Subsequent steps remain unchanged. This is motivated by the high amount of noise, especially for samples within B. Hence, the helper-task, that should focus on the relation between w_1 and w_2, may not depend on dimensions of w_1 in the

[61] α for 10 iterations: $\alpha = [0.5, 0.5, 0.5, 0.5, 0.5, 0.75, 1.0, 1.0, 1.0, 1.0]$

[62] α for 10 iterations: $\alpha = [0.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1.0]$

[63] α for 10 iterations: $\alpha = [1.0, 1.0, 1.0, 0.5, 0.0, 0.25, 0.5, 1.0, 1.0, 1.0]$

sentence representation, but on arbitrary other dimensions. For this particular case we do leave B untouched (and do not calculate it as $B = B \setminus (S \cup A)$), since the conflict is resolved by focusing on different parts of the sentence-representation. We refer to `(max-pool)`, wen applying this strategy.

6.2 Extraction of WordNet data

We experiment with different strategies how to extract relevant data from WordNet, considering the lexical semantic relations, as described in Section §2.2, as relevant. We find, that by aiming for a high recall, thus considering *all* senses of each word, a lot of noise is added, especially for co-hyponyms. For instance consider the word "blue", which is in almost all cases mutually exclusive to other colors like "red", "green" or "yellow", thus we intend to find these colors. While this can be extracted within the first synset, blue contains 16 different senses, ranging from "sky", " amobarbital sodium" or a "family of butterflies" to adjectives like "aristrocratic" or "depressed", yielding in antonyms of "blue" like "lowborn", "cheerful" or even "clean" and synonyms as according to oher words for each sense. The impact is even stronger, when identifying co-hyponyms. As seen in Section §3.1.1, WordNet maintains in many, but not in all cases, a very fine-grained hypernomy graph, requiring us to not only consider the next hypernym, but several hypernyms along the path. Doing so, words like "miller" (a type of a moth) are considered as co-hyponym, naturally in an increasing frequency, as co-hyponyms in a tree-like structure appear exorbitantly more often than antonyms or synonyms.

6.2.1 Strategy to extract data

Instead of the previously explained approach, we only consider the first synset of each word and detect related synsets via the lexical relations. This does not remove noise, yet reduces it tremendeously, at the cost of neglecting several useful lexical semantic relations, or even using the majorily wrong sense[64]. The main motivation is, that we lack of automatic evaluation emthods for different types of extracted data w.r.t. SNLI, conflicting with the time constraints for the remainder of the work. Since previous experiments[65] focused on a high recall and did not improve the performance, we go in the opposite dirtection, aiming for a high precision of the extracted data. Note, that the WordNet baseline, as defined in Section §5 did not suffer from the same problem, since all word-pairs (w_p, w_h) already are known to have a meaningful relation (based on their creation process), thus the extracted relation between both words most likely is valid. The problem only occurs, as we intend to extract words the other way around, by knowing the relation and finding appropriate words.

6.2.2 Final extracted data

Specifically, by only looking at the first senses of each word w, we consider all lemmata within the synset as synonyms of w. Hypernyms of w are considered up to an edge length of 5. For co-hyponyms, we consider all hyponyms of hypernyms of w, both bound to an edge length of three, if and only if they are not also a hypernym of w. We also consider *part-meronyms*, of synsets, that are any kind of hyponym of "location". Since the interpretation of meronyms is not trivial w.r.t. entailment, we thus only consider it in the context of locations and assume that a meronym entails its holonym[66].

A (entailed)		**B (contradicting)**	
$\mathbf{w_1}$	$\mathbf{w_2}$	$\mathbf{w_1}$	$\mathbf{w_2}$
oppose	contend	Trojan	Iraqi
pug	dog	waffle	Cheesecake
reward	rewarding	five	trio
townspeople	town	inferno	radius
pop	bulge	conditioner	aerobics
permit	permit	killers	party
frolics	play	hiding	processes
leading	ahead	chapel	synagogue
commitment	sincerity	Villages	crossroads
Wool	material	saloon	Minivan

Table 19: Examples of extracted word-pairs (w_1,w_2) for both categories, being represented by the sentence containing w_1 (thus A) or not (thus B).

[64] In Section §3.1.1 we showed that table (in the sense of a tabular visualization) is the first sense of table (as opposed to the sense of a furniture). Yet, in SNLI we especially expect the second sense to be useful.

[65] Conducted by Vered Shwartz and not part of this work.

[66] As in "John is in Berlin." ⇒ "John is in Germany.", with "Berling" *part-of* "Germany"

Using the explained extraction method, we generate a total of 686,265 word pairs with their entailment interpretation into either A or B, precisely $|A| = 104,550$ and $|B| = 581,715$. Note that these include the word itself like (w_1,w_1, A), if lexical semantic relations for a word are found. All words appear within the SNLI dataset and thus can be useful. We show a random samples of ten pairs for each class respectively in Table 19. Obviously, the data is still not entirely clean, however one can at least identify, why several word-pairs are within A or B respectively. Applying the extracted data on sentences within SNLI train, yields to an average of 44.6 samples from A and 300.0 samples from B for each single sentence, p or h, for the helper-task.

6.3 Evaluation

We evaluate all previously explained experiments within this section.

6.3.1 Integrate WordNet using embeddings

Table 20 shows the performance of the concatenated word-embeddings together with the performance of the unchanged Residual-Stacked Encoder†. The upper part contains embeddings that have been newly created or changed to contain other than (only) distributional information, the lower part shows the concatenated hypernyms, using the original (but fine-tuned during training) distributional representations. All methods slightly decrease in terms of accuracy for the original SNLI test set, even though only additional information are added. This however is not significant and most likely stems the parameters being highly tuned towards SNLI for the original model. Since we do not intend to increase the performance by a small margin coming from hyperparameter settings, we did not fine-tune these parameters. Also on the new test set, most approaches do not show major differences to the original model. Only the concatenation of hypernyms with an increasing amount shows an improvement, which is even stronger than the accuracy achieved by ESIM (Chen et al., 2017b).

Additional Embeddings	**Dimensions**	**SNLI test**	**Δ**	**New test**	**Δ**
Attract-Repel (Rücklé et al., 2018)	300D	85.4%	−0.4	58.3%	−0.9
Trained-All (cross-entropy)	50D	85.4%	−0.4	59.2%	±0
Trained-Syn-Ant (eucledian)	20D	85.4%	−0.4	57.8%	−1.4
Trained-Syn-Cohyp (eucledian)	20D	85.5%	−0.3	55.7%	−3.5
Trained-Syn-Cohyp-Ant (eucledian)	20D+20D	85.7%	−0.1	56.6%	−2.6
Hypernyms-1	300D	85.2%	−0.6	54.8%	−4.4
Hypernyms-3	300D	85.3%	−0.5	60.8%	+1.6
Hypernyms-5	300D	85.4%	−0.4	66.4%	+7.2
Residual-Stacked Encoder†	–	85.8%	±0	59.2%	±0

Table 20: Evaluation of experiments with additional information in the word-representations, compared to the Residual-Stacked Encoder† (bottom).

6.3.2 Integrate WordNet using multitask-learning

We depict the results of experiments using multitask-learning as descibed in Section 6.1.3, in Table 21. The first column shows the dimensions of the helper-task MLP. We only used two layer MLPs with relatively many dimensions, as previous experiments showed that smaller networks have already problems reaching a high accuracy of the helper-task. The presented networks all solve this task with an accuracy of $> 90\%$ (dev). Similarily, we observe that, reducing α, thus increasing the impact of the helper-task, generally reduces the performance. This can be seen in the upper part of the table. Generally, reducing the impact of the helper-task, by increasing the complexity of the helper-task MLP, omitting dropout or by increasing α, the performance drops less (or slightly improves). Subsequentially, experiments with a very strong impact of the helper task, as sharing a layer of its MLP or freezing its weights (third part of the table), reflect in a very poor performance. Similarily for the dynamic adaptions for α in the second part of the table only show comparable results to the original model, if the helper-task only is considered for a few iterations, with the model tuning to SNLI in the begiinning and end of the training. By taking max-pooling information into account, the performance only decreases slightly. While this potentially should lay the focus on relevant words, we eglect the fact that certain attributes may be shifted, due to the context implementing nature of LSTMs and indicated by the slight noisyness of dimensions. Thus, for “a happy child” the information for *being happy* might not only be present within “happy”, but also within “child”, as the first describes the second word. This may result in “child” having a higher value within the relevant dimension. While this dimension may still be relatively high for “happy”, in our experiment we neglect this dimension completely

Helper-task MLP	α	dropout	Re-sample less frequent	SNLI test	Δ	New test	Δ
2 × 100D	0.5	–	yes	84.6%	−1.2	47.2%	−12.0
2 × 600D	0.5	–	yes	85.2%	−0.6	59.9%	+0.7
2 × 300D	0.5	yes	yes	84.8%	−1.0	52.5%	−6.7
2 × 600D	0.5	yes	yes	84.8%	−1.0	51.8%	−7.4
2 × 300D	0.5	yes	–	85.2%	−0.6	48.7%	−10.5
2 × 600D	0.5	yes	–	85.0%	−0.8	57.7%	−1.5
2 × 300D	0.75	yes	yes	85.3%	−0.5	61.0%	+1.8
2 × 300D	0.75	yes	–	85.7%	−0.1	58.9%	−0.3
2 × 300D	`finetune`	yes	yes	84.9%	−0.9	51.5%	−7.7
2 × 300D	`finetune`	–	yes	84.5%	−1.3	52.3%	−6.9
2 × 600D	`focus-start`	–	yes	85.4%	−0.4	46.9%	−12.3
2 × 600D	`focus-mid`	–	yes	85.4%	−0.4	59.6%	+0.4
2 × 300D (freeze)	0.75	–	yes	85.3%	−0.5	53.2%	−6.0
2 × 300D (freeze)	0.5	–	yes	84.7%	−1.1	44.8%	−14.4
2 × 800D (shared)	0.5	yes	yes	84.4%	−1.4	42.2%	−17.0
2 × 600D (shared)	0.75	yes	yes	84.6%	−1.2	42.3%	−16.9
2 × 400D (shared)	0.5	yes	yes	83.9%	−1.9	34.3%	−24.9
2 × 300D (max-pool)	0.75	yes	yes	84.8%	−1.0	57.8%	−1.4
Residual-Stacked Encoder†	–	–	–	85.8%	±0	59.2%	±0

Table 21: Evaluation of experiments using multitask-learning, compared with the Residual-Stacked Encoder†.

(by setting it to zero). Thus a more sophisticated approach might be to use the output vectors after each timestep for the according word directly, instead of masking the final sentence representation.

Re-sampling the helper task such that $|A| = |B|$ seems superior to not-resampling. Not all evaluated models are completely comparable, as multiple parameters may differ. This is due to no findings with promising results, yielding in broadly covering experiments. Except for some small insignificant improvements on the new dataset, no experiment showed any success.

6.4 Analysis

We analyse selected experiments of both approaches, using concatenated emnbeddings or multitask-learning in this section, showing their strengths and disadvantages.

6.4.1 Integrate WordNet using embeddings

Table 22 shows the accuracy per category for `Attract-Repel`, as being the most sophisticated word-representations additional non-distributional information and `Hypernyms-5`, as the best achieved result. Comparing the model with concatenated embeddings from Attract-Repel with Residual-Stacked Encoder† shows, that different features are considered relevant. Most of the differences within the categories seem arbitrary, arsing from this issue, as the original information would still be accessable to the model. Synonyms are strongly improved compared with the original model, which can easily be explained due to an even higher word-vector similarity from Attract-Repel post-processed representations. On the other hand, both antonym groups show no big improvement, the antonyms derived from WordNet show an even worse accuracy than before. Overall, the performance gained using these embeddings with all deviations within different categories is highly similar to the ones achieved using multitask learning (next Section). Since the added word-vectors encode differences for most antonyms, but do not leverage them, we assume this stems from a lack of representative data within SNLI data. Naturally, if the model does not depend on those differences during training, it will not learn to consider them in the prediction process.

Impact of hypernyms

Looking at the concatenation of the mean of five hypernyms for each word, the increase in performance looks much more stable. Ignoring *planets*, which are highly noise-sensitive due to their limited size, only two categories do not improve over the baseline. Note, that antonyms include word-pairs like “day” and “night” and hyperthetically may share the same

Category	Amount	Residual-Stacked Encoder†	Attract-Repel	Δ	Hypernyms-5	Δ
antonyms	1,147	51.0%	53.8%	+2.8	74.2%	+23.2
cardinals	759	20.3%	19.1%	−1.2	15.3%	−5.0
nationalities	755	44.2%	31.1%	−13.1	56.7%	+12.5
drinks	731	89.7%	93.3%	+3.6	72.7%	−17.0
antonyms(WN)	706	63.2%	60.8%	−2.4	71.2%	+8.0
colors	699	90.8%	88.1%	−2.7	95.2%	+7.1
ordinals	663	3.0%	8.8%	+5.8	6.2%	+3.2
countries	613	75.4%	44.1%	−31.3	81.6%	+6.2
rooms	595	73.1%	77.0%	+3.9	75.0%	+1.9
materials	397	80.4%	77.6%	−2.8	85.1%	+4.7
vegetables	109	40.4%	41.3%	+0.9	41.3%	+0.9
instruments	65	96.9%	98.5%	+1.6	98.5%	+1.6
planets	60	61.7%	31.7%	−30.0	38.3%	−23.4
synonyms	894	73.9%	92.6%	+18.7	74.9%	+1.0
total	8,193	59.2%	58.3%	−0.9	66.4%	+7.2

Table 22: Accuracy per category for concatenated embeddings using Attract-Repel or Hyponyms-5.

hypernyms, as well as co-hyponyms do. Since, synonyms obviously also share similar hypernyms, the model must to learn this differentiation. Looking at the categorical evaluation, the performance of synonyms remains similar to the original evaluated model, showing that it does not suffer from added hypernyms in this aspect. Yet, we observe that adding hypernyms does not help in identifying synonyms, which is based on the results in Section §5 a relatively easy do classify category. Especially the overall improvement for contradicting examples is interesting and we closer examine the impact of the hypernyms. To focus only on this aspect, in this anaysis we exclude samples, that are predicted identically as by the Residual-Stacked Encoder†. Even though the concatenated embeddings may have influenced those, too, we thus exclude samples that are correctly classified by memorizing word-relations form SNLI. Figure 27 visualizes the impact of

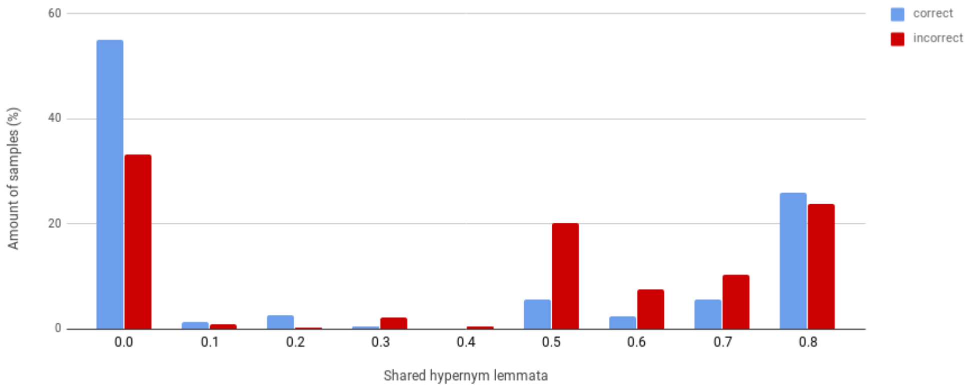


Figure 27: Comparison of contradicting samples (different w.r.t. correctness from Residual-Stacked Encoder†). for Hypernyms-5, by the amont of shared hypernym embeddings.

the hypernym embeddings by comparing how many hypernym embeddings are identicalw wor each word-pair(w_p,w_h). In total, 1378 contradicting samples are classified differently from the original model, 984 are now classified correctly as contradiction (blue), 394 samples are now misclassified (red). The x-axis shows the percentage x_{w_p,w_h} of shared hypernym lemmata between w_p and w_h, calculated as

$$x_{w_p,w_h} = \frac{|H_{w_p} \cap H_{w_h}|}{|H_{w_p} \cup H_{w_h}|} \tag{17}$$

with H_{w_p} and H_{w_h} being the sets lemmata of hypernyms from w_p and w_h respectively, gathered using the same method as to create the embeddings. The y-axis displays the proportinal amount of each group. Contrarily to our intention, the model does not strongly use the additional vectors to identify co-hyponyms. Instead, especially if only few hypernyms are identical, more samples are re-predicted correctly, whereas a highr similarity of the hypernyms leads to a higher amount of incorrect predictions. unrelated words being an indicator for contradiction highly correlates with the data seen during training. Due to the creation process of SNLI (in addtion to to frequently changed words as identified by Gururangan et al. (2018)) many contradicting samples describe very unrelated scenarios, yielding in contradiction based on the event-coreference, not because they contain related, but contradicting, words (Dasgupta et al., 2018). Subsequentially, unrelatedness of words serves as a good indicator for contradiction. Yet, looking at different categories, we observe that at least for some of them, the model learned the tendency of leveraging from the added information in the intended way. Cardinals in general all share ≥ 0.7 % of their hypernym lemmata, thus are very similar to each other. A total of 200 cardinal samples are either in the red or blue group visualized. Of those, 143 (71.5%) have been classified correctly with the concatenated hypernyms and only 57 (28.5%) are misclassified compared to the original predictions. Similar, but less strongly, are 63 (61.8%) countries with ≥ 0.7% shared hypernym lemmata classified correctly and only 39 (38.2%) incorrectly. Opposed to these categories, for nationalities or antonyms, the improvements compared to the base model arises from unrelated word-pairs (in terms of their shared hypernym lemmata). In both cases, the majority of different re-prediction stems from word-pairs sharing no lemma within their hypernyms. Only looking at those word-pairs, sharing not a single word-vector for their hypernyms, 109 (88.6%) of 123 nationalities[67] and 278 (90.3%) of 308 antonyms[68] are re-predicted correctly. In total, the model seems to slightly benefit from the added information in some cases in the intended way, in the majority of cases however, the improvement in performance seems to stem from another often occuring pattern in SNLI, namely unrelated hypothesis, rather than improving the general NLU.

6.4.2 Integrate WordNet using multitask-learning

We select for the analysis the best performing model on the new test-set using a 300-dimensional helper task MLP with $\alpha = 0.75$ with re-sampling and dropout, denoted as `300D-0.75 STD` and the comparable model with with a stronger impact of the helper-task with $\alpha = 0.5$, also 300 dimensions, resampling and dropout, named as `300D-0.5 STD`. Additionally we select the model using the max-pooled information, referred to as `300D-0.75 max-pool`, as it puts the focus on the actual word relations and performs comparably with the Residual-Stacked Encoder†. The accuracy per category of these models is displayed in Table 23. It can be seen that the majority of all contradicting categories performs worse than the base-model. Only the synonyms highly leverage from this method and assemble the performance, reached by attention-based models in Section §5. We observe this effect on synonyms for all 17 evaluated models of Section §6.3.2. The extracted synonyms from WordNet are much less noisy than co-hyponyms, which may lead to the interpretation, that the helper-task is more likely to consider the relevant dimensions coming from the synonym for its prediction. However, as seen by the model, taking max-pooling information into account, the performance on contradicting samples is still very

Category	Amount	300D-0.75	Δ	300D-0.5	Δ	300D-0.75	Δ
		STD		STD		max-pool	
antonyms	1,147	39.4%	−11.6	35.6%	−15.5	44.9%	−6.1
cardinals	759	40.1%	+19.8	26.2%	+5.9	33.3%	+13.0
nationalities	755	52.5%	+8.3	32.1%	−12.1	25.3%	−18.9
drinks	731	75.9%	−13.8	67.4%	−22.3	82.5%	−7.2
antonyms(WN)	706	60.5%	−2.7	56.2%	−7.0	56.5%	−6.7
colors	699	90.6%	−0.2	86.3%	−4.6	88.0%	−2.8
ordinals	663	3.0%	±0	2.7%	−0.3	3.2%	+0.2
countries	613	69.8%	−5.6	40.6%	−34.8	69.7%	−5.7
rooms	595	74.1%	+1.0	65.2%	−7.9	74.6%	+1.5
materials	397	85.9%	+5.5	79.8%	−0.6	72.3%	−8.1
vegetables	109	41.3%	+0.9	44.0%	+3.6	32.1%	−8.3
instruments	65	95.4%	−1.2	93.8%	−3.1	95.4%	−1.5
planets	60	35.0%	−26.7	26.7%	−35.0	58.3%	−3.4
synonyms	894	97.6%	+23.7	96.1%	+22.2	94.4%	+19.5
total	8,193	61.0%	+1.8	52.5%	−6.7	57.8%	−1.4

Table 23: Accuracy per category for selected models using multitask-learning.

[67] Other **mispredictions**: 2× with 0.8 shared, 3× with 0.3 shared. Other **correct** predictions: 5× with 0.1—0.5 shared, 16× with 0.7—0.9 shared

[68] Other **mispredictions**: 52 × 0.5—0.9 shared. Other **correct** predictions: 5× with 0.2—0.5 shared, 41 with 0.5—0.9 shared.

poor. Only two categories seem to be improved on a relatively constant basis. Cardinals improve in 7/17 approaches, mostly by more 10 ten points in accuracy, vegetables improve on 9/17 approaches, with a maximum of 9.1% increase. Especially the drop within the second model for countries is severe, yet not only present within this model. Not a single model of our evaluations superceeded the original model for countries, 8/16 decreaed by more than 35 points in accuracy, another 2 by more than 10 points. Since the other models of Section §5 achieve similar results, and Residual-Stacked Encoder† is highly aligned with its hyperparameters to Residual-Stacked Encoder$^{\Diamond}$, we assume the high performance arises from correctly picked features by chance, rather than stemming from the model's architecture.

Impact of the selected data

Due to our restricted word-pair extraction method, defined in Section §6.2.1, the upper bound, that can be achieved using this information drops. Thus, in the following analysis we only look at samples, that could have been classified correctly, based on the extracted data from WordNet. The results are depicted in Table 24. We report the absoolute amount of samples together with the percentage of the original size of each category. The categories drinks (3), vegetables (11), planets (20), materials (35) and planets (37) are aggregated due to insufficient amount of samples to derive any representative conclusions. All Δ show the difference to the Residual-Stacked Encoder† on the same data, instead of comparing them with the same model using the full data. The 100% accuracy is not very suprising, given the fact that only synonym examples are included, if they can be explained using the extracted data. Thus, samples of this category that have another label than entailment, due to the usage in context, are excluded. Ther performance gain in the aggegated *other* category mostly stems from materials. Even though the models perform higher on some categories, compared to the performance over all data, except for cardinals and synonyms, all are generally predicted worse than by the base model without multitask-learning. If we compare the overall performance of each model with the performance achieved over all data, we observe that only the original Residual-Stacked Encoder† improved in accuracy, while other models perform worse than before. Subsequentially, they perform slightly better on the other half of the data, that cannot be explained using the fused information. Chen et al. (2017a) show with KIM extracted a total of 5,425,426 word pairs[69], being crucially more than ours (686,265), that the network especially levergaes from those, if ≥ 40% of the external knowledge is used. Compared too that, our experiments may indeed suffer from limited coverage, especially since Chen et al. (2017a) directly encode the lexical relations and we still rely on the MLP to leverage from them, indirectly encoded within the sentence-representation. Yet, it could have been expected, that models would have an advantage on this subset of data. Since they obviously fail to take advantage of the fused information, sufficient for all those samples, the problem seems to rather be the method than the data.

	Amount		**Residual-Stacked Encoder†**	**300D-0.75 STD**		**300D-0.5 STD**		**300D-0.75 max-pool**	
Category	#	%	Acc.	Acc.	Δ	Acc.	Δ	Acc.	Δ
antonyms	885	77.2%	51.3%	37.7%	−13.6	36.5%	−14.8	46.6%	−4.7
cardinals	496	65.3%	21.6%	41.2%	+19.6	29.6%	+8.0	34.7%	+13.1
countries	471	76.8%	75.6%	66.2%	−9.4	33.3%	−42.3	68.2%	−7.4
nationalities	427	56.6%	43.4%	59.3%	+15.9	32.8%	−10.6	32.1%	−11.3
antonyms(WN)	379	53.7%	77.6%	76.0%	−1.6	70.8%	−6.8	69.7%	−7.9
colors	312	44.6%	95.8%	95.8%	±0	93.3%	−2.5	93.3%	−2.5
ordinals	263	39.7%	6.5%	7.2%	+0.7	6.5%	±0	6.5%	±0
rooms	213	35.8%	94.8%	85.4%	−9.4	76.1%	−18.7	95.3%	+0.5
other	106	7.8%	33.0%	48.0%	+15.0	52.8%	+19.8	33.0%	±0
synonyms	385	43.1%	98.2%	100.0%	+1.8	99.7%	+1.5	100%	+1.8
Total	3,937	48.1%	60.1%	59.5%	−0.6	49.7%	−10.4	57.5%	−2.6

Table 24: Accuracy per category of three selected multitask-learning experiments compared with Residual-Stacked Encoder† on samples covered by extracted word-pairs.

Impact on the sentence representation

Looking back to the original intention, of fine-tuning embeddings and esuring those differences would be present within the sentence-representation, we take a closer look at this aspect. Following the results from the previous step we can

[69] Note, that they have a align words of p and h to identify their relation, thus they may not suffer that much from arbitrary word relations, extracted from WordNet.

only see, that the multitask-learning is not heelpful for the final prediction, which can have several reasons: Either the helper-task did not manage to encode the relevant differences into the sentence-representation, or it did, but the model failed to take advntage of them. We us the same technique as introduced in Section §4.3.2, visualizing the alignment of the sentence-representations of p and h. To only focus on samples with the required knowledge fused, we only calculate the averaged counts for the same subset of the data from the previous section, but only looking at contradicting samples (3511). Figure 28 shows the aligned sentences for the original Residual-Stacked Encoder†, differentiating between the 1954 correctly classified samples and 1557 mis-classified samples. As expected the majority of dimensions have the same value, arising from the high lexical overlap and the fact that word-pairs are selecterd to be replacable in context, thus will have similar embeddings. Even though the network structure, dimensions and performance, compared to the analysed model Shortcut-Stacked Encoder†, slightly changed, the results gained from this section still seem applicable. Due to the similarity of both sentences, only few dimensions differ. Taking this into consideration we observe that more of these differing dimensions for the correctly classified samples, in many cases morw than twice the amount of differeing dimensions compared to the misclassified examples. Knowing that the Residual-Stacked Encoder† most likely follows the same principles as identfified in Section §4, we now look at the sentence-representations gained from multitasking (for the same data). Figure 29 shows the aligned sentence-representations (correct and mis-predicted) for the best multitask-learned model *300D-0.75 ST*. Comparing these visuaizations with the orignal model, clearly both, the mispredicted and correctly predicted sentence-representations show a higher amount of different high-valued dimensions. In line with our

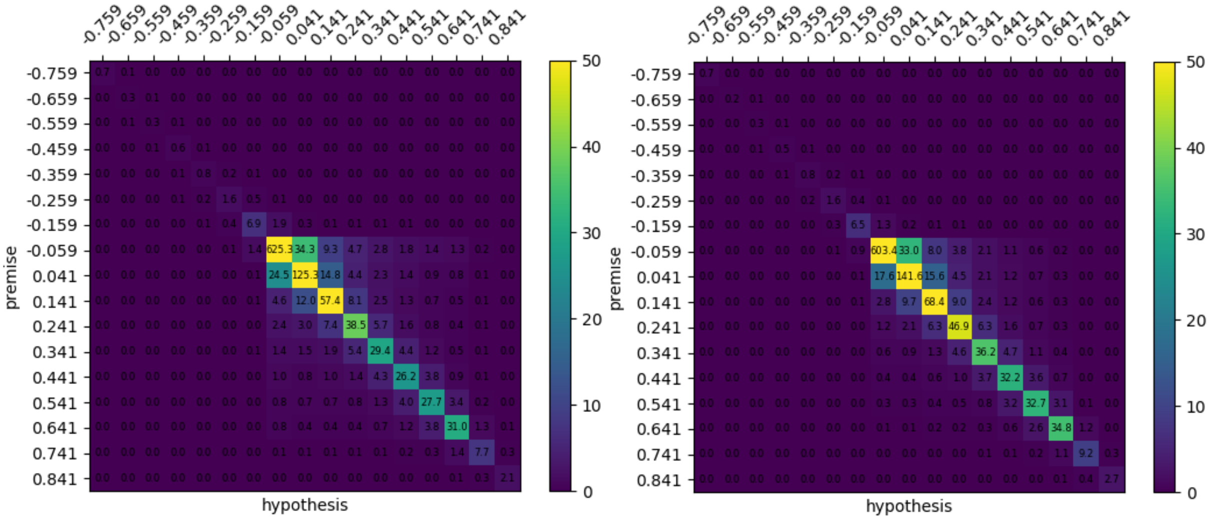


Figure 28: Aligned p and h for all contradicting sampes, covered by the fused WordNet information, correctly predicted (left) or mis-predicted (right).

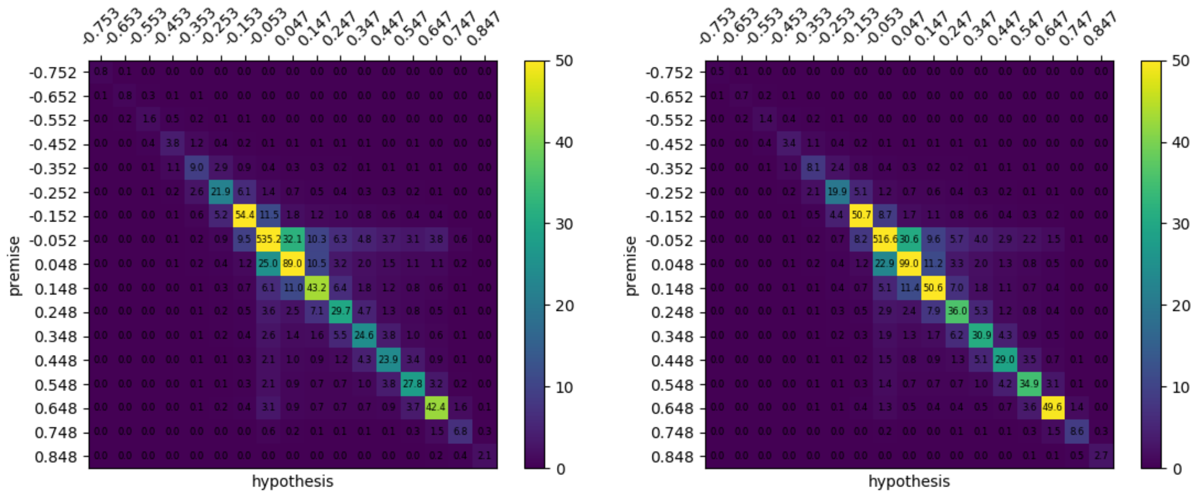


Figure 29: Aligned p and h, correctly predicted (left) and mis-predicted (right) for multitask-learned *300D-0.75 STD*.

other observations, this is especially visible for the correctly classified samples. While the Residual-Stacked Encoder†, as well as the Shortcut-Stacked Encoder† showed the majotity of dimensions within the positive values area, in this case a huge amount of samples are also within the negative area. We do not start another dimension-wise analysis for this model and leave it open for interpretation. Since this phenomen stems from the helper-task, one possible explanation is, that low values indicate the absence of specific words. Having a large lexical overlap, p and h lack the same words from B for the majority of words, thus the dominant symmetry for the negative values may arise. Even though the main-task may still optimize to deal with negative values or interpret the absence of information not aligned with zero but another value, this breaks the same behaviour that was naturally learned by both models without multitask-learning Similarly, XX showed, that even though the model are able to learn element-wise difference and product (which both work intuitively well with absence of information being encoded close to zero) by themselves, using it in the feature concatenation helps the performance. Yet, we do not further investiate this phenomena and it may not even be harmful to the final prediction. Similar results are shown for the other two experiments, picked for the analysis part[70]. Figure **??** shows all contradicting and entailing samples of the selected data, regardless of their prediction of the model, encoded by *300D-0.75 max-pool*. This seems to be a bit more fuzzy, compared to the other two multi-tasking experiments. Neglecting the high amount of negative valued dimensions, this is exctly the kind of representations we were aiming for, clearly encoding the same information fr entailing examples, while also encoding differences for the contradicting samples. As seen in Section §4.3.4, this however is not sufficient for the model to predict contradiction, as distinct information is also present for neutral samples. We conclude, that we managed to shift the sentence-representation in a way, that it encodes differences for antonyms and co-hyponyms stronger than before. However the classifying MLP lacks to leverage from those in the intended way. The many negative valued dimensions may also indicate, that the helper-task and the main task put more emphasis on different dimensions.

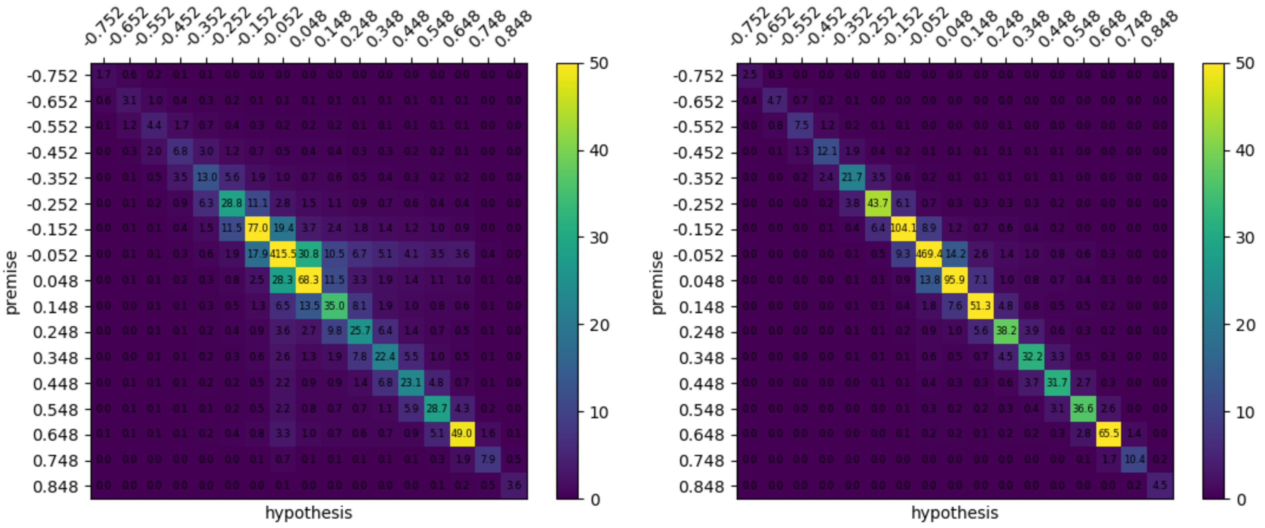


Figure 30: Aligned p and h, of contradicting (left) and entailing (right) samples for multitask-learned *300D-0.75 max-pool*.

6.5 Summarizing experiments to incorporate WordNet

We encountered the challenge of gaining high quality data out of WordNet. Even though lexical resources contain a huge amount of information, it is requred to put more effort into the extraction of the data. In our case, we applied a simple strategy to increase the precision of the exracted data, yet at the expense of a lot of valuable information. We have shown, that adding additional information to the network inputs (the embeddings) may have a good impact in some cases, however we still rely on the train data to rely on those additional information and the model to identify and consider them as relevant features. This is somewhat challenging, since some of these information might be less frequent in the data, while highly represented arbitrary patterns within the train data are more important w.r.t. the optimmization function. We tackled this problem using multitask-learning and have successfully transferred word similarities based on the fused WordNet information into the sentence-representation, yet the final MLP failed to consider them in the desired way. This basically breaks down to the same problem, that the changed encodings are not considered relevant w.r.t. the train data.

[70] We did not conduct similar analysis for the remaining models.

Comparing to KIM

As opposed to the sentence-encoding Residual-Stacked Encoder, KIM (Chen et al., 2018) uses inter-sentence attention and identifies directly the WordNet relation for words within p and h, while we only use a indirect way to encode this. Theit approach however seems very elegant, if considering the heavy influence of the train data, whether features are considered relevant or not. Gururangan et al. (2018) identified the SNLI-specific patterns not as a problem because they are incorrect, but because they are highly dominant, leading to oversimplified solutions only based on those. Thus, a model has a good chance of being correct to classify a sample as contradicting if a "cat" is in the h and a "dog" is in p, which is not challenging to learn for a neural network if seen in a large number of times. However, instead of memorizing this specific word-pair, knowing that both words are close hyponyms of the same hypernym ("pet") also serves a simple and effective feature. By assigning each lexical relation (quantified by their distance) holding between two words, onto one specific dimension, Chen et al. (2018) made this simple feature easily accessable and also show that especially the information for co-hyponomy is beneficial. Subsequentially to having this *meaningful* decision criteria, the model can apply the same strategy on other co-hyponyms and indeed achieve a better generalization ability. As opposed to their strategy, using the indirectd way to encode differences still relied on the importance of those difference for the train data. Even if the same lexical relation can be inferred from those new representations (which would be the best case, but most likely is not the case), this may depend on different dimensions, depending on the actual words. We conclude that WordNet information must be fused in a way, that is generally applicable and can easily be identified by the model, in order to overcome certain patterns and be more useful than memorizing.

7 Conclusion and future work

In this work we showed at the sampe of SNLI, that even though being intended to improve NLU, the high performance gained by state-of-the-art models for NLI does not reflect the actual NLU capabilities. All models performed significantly lower on our adversarial dataset, derived from the original train-set but excluded SNLI-specific patterns. Even though KIM performed quite well, future work may find ways to integrate more data or methods to deal with lexical inferences in context to improve the performance, which has not yet met the limit. We attempted to improvce the performance on this new dataset using WordNet information for a sentence-encoding model. Here, we leveraged from the max-pooled sentence-encoding and showed that this can be used to understand the dimensional values and sentence-representations, generated by the model. In addition to showing that this information can indeed be used to change the representations in a meaningful way, our inisghts gained here, have also shown to be useful for an intrinsic analysis of the sentence-representations of the experiments incorporating WordNet. Future work can develop deper insights on these representations on a broader range of data and models, to enable more meaningful analysis or even adaptions. Finally we evaluated in our approaches to incorporate WordNet, that this should be done in a very easy to identify way, in order to be relevant enough to overcome diminant patterns in a dataset. Especially for sentence-encoding models this seems very challenging at the moment, future work may leverage from structures like memory networks (Sukhbaatar et al., 2015), to do so.

8 Acknowledgements

Special Thanks to Yoav Goldberg and Vered Shwartz

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, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

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, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(4):i–xvii.

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.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

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*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

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.

Çağlar Gülçehre and Yoshua Bengio. 2016. Knowledge matters: Importance of prior information for optimization. *The Journal of Machine Learning Research*, 17(1):226–257.

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*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

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*.

Germán Kruszewski and Marco Baroni. 2015. So similar and yet incompatible: Toward the automated identification of semantically compatible words. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 964–969.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.

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.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

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. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

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. *RepEval 2017*, page 1.

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

Patrick Pantel. 2005. Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 125–132. Association for Computational Linguistics.

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*.

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*.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

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*.

Vered Shwartz and Ido Dagan. 2016. Adding context to semantic data-driven paraphrasing. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 108–113.

Vered Shwartz, Omer Levy, Ido Dagan, and Jacob Goldberger. 2015. Learning to exploit structured resources for lexical inference. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 175–184.

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.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

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