

Deep Learning for Natural Language Inference: A Literature Review

Aurélien Coet

University of Geneva
Faculty of Sciences,
Dpt. of Computer Science

March 2019



**UNIVERSITÉ
DE GENÈVE**

FACULTÉ DES SCIENCES

Contents

1	Introduction	1
1.1	Definition and Scope of NLI	1
1.2	Interest and Applications	3
1.3	Contents of the Document	4
2	NLI Tasks and Data Sets	5
2.1	The RTE Challenges	5
2.2	The SICK Data Set	6
2.3	The SNLI Corpus	7
2.4	The MultiNLI Corpus	9
2.5	The Breaking NLI Data Set	11
2.6	Other resources	12
3	Deep Learning Models for NLI	13
3.1	Sentence Vector-Based Models	13
3.1.1	Classification Layer	14
3.1.2	Sequential Encoders	14
3.1.3	Tree-based Encoders	16
3.1.4	Self Attention-based Encoders	17
3.2	Sentence Matching Models	19
3.3	Transfer Learning Approaches	23
3.3.1	Transfer Learning from Supervised Tasks	24
3.3.2	Transfer Learning from Unsupervised Tasks	25
4	Conclusion	29

List of Figures

3.1	Representation of the Siamese architecture for NLI	14
3.2	Classifier architecture in sentence vector-based models	14
3.3	k-layers stacked bi-LSTMs with shortcut connections for sentence encoding	15
3.4	Inner attention mechanism (Y. Liu et al., 2016)	18
3.5	Word-by-word attention mechanism (Rocktäschel et al., 2015)	20
3.6	Attend-compare-aggregate architecture (Parikh et al., 2016)	21
3.7	ESIM architecture (Chen, Zhu, Ling, Wei, et al., 2017a)	22
3.8	DRCN architecture (Kim et al., 2018)	23
3.9	Transfer learning in CoVe (McCann et al., 2017)	25
3.10	Architecture of the GPT and transfer to other tasks (Radford et al., 2018)	27
3.11	Transfer learning to NLI with BERT (Devlin et al., 2018)	28

List of Tables

1.1	Examples of sentence pairs and their associated labels	2
2.1	Examples of sentence pairs from RTE-1 (Dagan et al., 2006)	6
2.2	Examples of sentence pairs from SICK (Marco Marelli et al., 2014)	7
2.3	Examples of instances from the SNLI corpus (S. Bowman et al., 2015)	8
2.4	Examples of instances from the MultiNLI corpus (Williams et al., 2018)	10
2.5	Examples of instances from the <i>Breaking NLI</i> data set (Glockner et al., 2018)	11
3.1	Reported accuracy (%) of sequential sentence encoders on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets	16
3.2	Reported accuracy of tree-based sentence encoders on SNLI’s test set	17
3.3	Reported accuracy (%) of self attention-based sentence encoders on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets	19
3.4	Reported accuracy (%) of sentence matching models on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets	24
3.5	Reported accuracy (%) of transfer learning approaches on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets	28

List of Acronyms

- AI** *Artificial Intelligence*. 1
- BERT** *Bidirectional Encoder Representations from Transformers*. ii, 27, 28
- bi-LSTM** *bidirectional LSTM*. ii, 15, 17, 18, 21, 22
- biLM** *bidirectional Language Model*. 26
- BiMPM** *Bilateral Multi-Perspective Matching*. 21
- c-LSTMs** *coupled-LSTMs*. 20
- CAFE** *Comprop Alignment-Factorised Encoders*. 21, 22
- CDSMs** *Compositional Distributional Semantic Models*. 6, 7
- CNN** *Convolutional Neural Network*. 16, 18
- CoVe** *Context Vectors*. ii, 24–26, 28
- DIIN** *Densely Interactive Inference Network*. 22
- DiSAN** *Directional Self Attention Network*. 18
- DL** *Deep Learning*. 4, 6
- DMAN** *Discourse Marker Augmented Network*. 25, 28
- DMP** *Discourse Marker Prediction*. 25
- DR-BiLSTM** *Dependant Reading Bidirectional LSTM*. 22
- DRCN** *Densely-connected Recurrent and Co-attentive neural Network*. ii, 22, 23
- ELMo** *Embeddings from Language Models*. 26, 28
- ESIM** *Enhanced Sequential Inference Model*. ii, 10, 21–23, 26
- GPT** *Generative Pre-training Transformer*. ii, 26–28
- GRU** *Gated Recurrent Unit*. 15
- IR** *Information Retrieval*. 5, 6

-
- KIM** *Knowledge-based Inference Model*. 23
- LMs** *Language Models*. 25, 26
- LSTM** *Long Short Term Memory*. 9, 15, 19, 20
- MLM** *Masked Language Model*. 27
- MLP** *Multi-Layer Perceptron*. 14, 21, 23
- mLSTM** *match-LSTM*. 20
- MT** *Machine Translation*. 24, 25
- MultiNLI** *Multi-Genre Natural Language Inference*. iii, 9–13, 16, 17, 19, 23, 24, 27–29
- NLI** *Natural Language Inference*. i, ii, 1–5, 9, 11, 13, 14, 17, 19, 20, 23–29
- NLP** *Natural Language Processing*. 1, 3, 17, 24–26, 29
- NLU** *Natural Language Understanding*. 1–4, 13, 27
- NNs** *Neural Networks*. 4, 7
- OANC** *Open American National Corpus*. 9, 10
- QA** *Question Answering*. 5, 6
- RC** *Reading Comprehension*. 5
- ReSAN** *Reinforced Self Attention Network*. 18
- rLSTM** *re-read LSTM*. 20
- RNN** *Recurrent Neural Network*. 14–16, 22
- RTE** *Recognising Textual Entailment*. 1, 2, 5–7, 9, 16, 25, 30
- SICK** *Sentences Involving Compositional Knowledge*. iii, 6, 7, 9, 12
- SNLI** *Stanford Natural Language Inference*. iii, 7–12, 16, 17, 19, 23–25, 27–29
- SPINN** *Stack-augmented Parser-Interpreter Neural Network*. 16, 17
- SRL** *Semantic Role Labelling*. 23
- TBCNN** *Tree-based Convolutional Neural Network*. 16, 17
- TreeRNNs** *Tree-structured Recursive Neural Networks*. 16, 17
- XNLI** *Cross-lingual Natural Language Inference*. 12
-

Chapter 1

Introduction

1.1 Definition and Scope of NLI

Natural Language Understanding (NLU) is a sub-domain of *Natural Language Processing* (NLP) and *Artificial Intelligence* (AI) which aims at giving computers the ability to understand and interpret human languages. In order to fulfill that goal, a number of difficult tasks involving syntactic and semantic aspects of natural language have been devised, such as *text summarisation*, *sentiment analysis* or *relation extraction*. Among these tasks is the central problem of *Natural Language Inference* (NLI), also known as *Recognising Textual Entailment* (RTE) (Dagan et al., 2006).

In NLI, the objective is to recognise whether a sentence p , called *premise*, entails another sentence h , called *hypothesis*. Here, we define *entailment* as the relation of information inclusion between the premise and hypothesis. That is, a sentence p entails h if and only if all of the information given in h is also present in p . In other words, a hypothesis is entailed by a premise if it can be inferred from it. An example is given below:

p: Two boys are playing football in a grass field.
h: Children are playing outside.

For a human reader, recognising that p entails h in the example above is done easily. Our brain is naturally able to determine that *two boys* are *children*, that a *grass field* is *outside*, and that *playing football* is summarised by *playing*. However, allowing a computer to perform the same task proves to be particularly hard. The difficulty of NLI comes not only from the complex nature and ambiguity of natural language, but also from the fact that the decision whether a sentence entails another is based on human judgement and not some kind of formal logic. Indeed, in NLI the relation linking a premise and hypothesis is usually chosen to be what someone would decide upon reading the sentences. Hence, giving a system the ability to detect entailment not only involves developing an understanding of the structure and meaning of language, but also reproducing the line of thought of humans.

If we consider the steps through which our brain went to recognise entailment in the previous example, a number of complex tasks both on the syntactic and semantic level can be identified. First, before the sub-parts of the two sentences can be compared, they must be identified and matched. It is hence necessary to parse the sentences into their subject, verb, object and complements, which implies having some kind of knowledge of English grammar. Once this is done, the semantic relations that lie between the matched phrases then need to be recognised and composed to make a decision about the sentence pair being observed. While entailment can easily be inferred from some of those relations (e.g. *playing football* obviously entails *playing*), it can also be very hard to infer from others (some advanced knowledge of the world is necessary to be able to determine that a *grass field* is located *outside*). Of course, the steps described above aren't always necessary for a system to perform well on NLI, but they illustrate the kind of challenges being faced when working on the problem.

So far, we have only defined NLI as a binary classification task in which a choice has to be made between entailment/no entailment based on a premise-hypothesis pair. In most formulations of NLI problems, however, the actual objective isn't to distinguish between two but three different classes: *entailment*, *contradiction* and *neutral* (S. Bowman et al., 2015; Conneau, Rinott, et al., 2018; Giampiccolo et al., 2008; Marelli et al., 2014; Williams et al., 2018). This particular kind of formulation makes the task at hand even harder, as not only information *inclusion*, but also information *exclusion* need to be recognised.

Sentence pair	Relation
p: A woman having a beverage. h: Woman has a drink.	entailment
p: Men are sitting at a table. h: People standing near a table.	contradiction
p: A man and his dog playing frisbee. h: A man is having fun with his dog.	neutral

Table 1.1: Examples of sentence pairs and their associated labels

Let's consider the three examples of sentence pairs and their associated labels in table 1.1. In order to properly classify them, a system must be able to identify that *beverage* and *drink* are synonyms in the first pair, that *standing* and *sitting* are antonyms in the second, and finally that *playing frisbee* doesn't necessarily imply *having fun* but doesn't contradict it either. Furthermore, while the sentences presented here do not illustrate it, premises and hypotheses in NLI tasks can sometimes contain grammatical errors and spelling mistakes, as they are usually written by humans. This kind of errors shouldn't prevent a system from correctly classifying instances, which only adds to the difficulty of the problem.

In their *MultiNLI* paper (Williams et al., 2018), the authors argue that the main difficulty in RTE is to extract meaningful representations for the sentences that compose the premises and hypotheses of an NLI dataset, which makes the task particularly interesting for representation and transfer learning. They also underline how the large variety of linguistic phenomena that must be handled by models to recognise entailment make it a good benchmark on NLU.

1.2 Interest and Applications

Considering all the elements mentioned in the previous section, one can easily imagine why natural language inference is such an important research interest in the field of NLP. The complexity of the task and its semantic relevance make it an essential aspect of NLU and a major problem that needs to be tackled in this area. The interest of NLI however goes beyond academic research, and numerous applications would benefit from the ability to perform inference on human language:

- In *automatic text summarisation* (Das and Martins, 2007), the objective is to automatically produce a summary for a piece of text or a document. The result should be short and remove any kind of redundancy from its source, but without losing any of the important information it contains.

In such a situation, NLI can for example be used to detect if any sentence of the summary can be inferred from the others (which would mean that it is redundant) (Dagan et al., 2006), or to verify that the summary is well entailed by the original text from which it was generated (to ensure that it doesn't include any additional or unrelated information) (Pasunuru et al., 2017). Inference can also be used to directly address the task at hand, by determining which sentences in the original document are entailed by other larger chunks of text and extracting them to build a summary.

- *Opinion summarisation* (Condori and Pardo, 2017) is a task that has seen a significant surge in interest over the past years, mainly due to the fast growth of social networks and online shopping websites. The idea in opinion summarisation is to analyse pieces of text written by different people on a specific subject (such as product reviews on Amazon or political opinions on Twitter) and to extract the general sentiments that are shared by multiple persons.

In this case, NLI can be used in a similar fashion as for automatic text summarisation: a sentence or a piece of text that is entailed by multiple opinions can be considered to summarise them well, as it contains no contradictory or additional information.

- In *reading comprehension* (Hirschman et al., 1999), a system takes some document as input and answers questions about it by searching for relevant information in the text.

In this type of problem, NLI can be used to find the sentences in the source text that can be inferred from the questions and using them to build answers. It can also serve to choose the best answer between potential solutions by determining which ones can be inferred from the questions with more confidence.

- *Question answering* is a task very similar to that of reading comprehension where the objective is to answer open domain questions by using diverse sources of information. In that situation, NLI can serve similar purposes as in reading comprehension.

1.3 Contents of the Document

Thanks to the relatively recent creation of large scale natural language inference data sets (more on that in chapter 2) and the fast development of *Neural Networks* (NNs) over the past few years, the NLU community has come up with numerous *Deep Learning* (DL) models to address the problem of NLI. This paper proposes a thorough investigation of such models.

The rest of this document is organised as follows:

- In chapter 2, we describe the most important tasks and data sets that have been created for the problem of natural language inference, and we underline the similarities and differences between them.
- In chapter 3, the most prominent deep learning models for NLI are presented, and their concepts are briefly explained and compared. The chapter is divided into three parts that correspond to different categories of DL models: sentence vector-based models, sentence matching models and transfer learning approaches.
- Finally, chapter 4 proposes a conclusion and final remarks on the subject of natural language inference, as well as potential directions for further research.

Chapter 2

NLI Tasks and Data Sets

Over the years, a number of tasks and datasets have been devised for the problem of natural language inference. In this chapter, we describe the most prominent ones and compare them.

2.1 The RTE Challenges

The PASCAL Network of Excellence *Recognising Textual Entailment* (RTE) challenge benchmark (Dagan et al., 2006) was the first to ever propose a unified task on which to evaluate NLI models. The initial instance of the challenge (RTE-1) was presented in 2005 during a workshop on textual entailment, and subsequent versions were proposed from 2006 to 2011 (RTE-2 to RTE-7).

The RTE-1 challenge consists in a series of sentence pairs called T for *text* and H for *hypothesis* and labelled with *entailment/no entailment*. The data is split in a development and a test set which contain 567 and 800 pairs, respectively. The objective of the challenge is to correctly predict, for each example in the test set, whether H is entailed by T or not.

For the definition of the data set, human annotators were asked to write hypotheses H that were entailed or not by given texts T . The task was defined such that a balanced proportion of entailment/no entailment examples were produced and that the pairs weren't too trivial to classify (for example, annotators were discouraged from producing hypotheses that had high word overlaps with the texts). Once the process of defining T/H pairs was over, cross-evaluation was performed on the data by asking each annotator to label the sentences produced by the others. Agreement was obtained in about 80% of cases, and all pairs for which no agreement was reached were discarded from the final data set.

One specificity of the RTE challenge is that its data set definition was grounded in a number of different applications from text processing. All the sentences corresponding to texts for which hypotheses had to be written were selected among typical examples encountered in tasks like *Information Retrieval* (IR), *Reading Comprehension* (RC) or *Question Answering* (QA), for example. The idea was that this would allow to compare the performances of systems built with different objectives in mind

on the shared task of predicting inference. Thanks to this, the sentences composing the pairs in the RTE data set often consist in real life examples of natural language and cover a wide range of topics. Examples of instances from RTE-1 (taken from the original paper) are presented in table 2.1.

Text	Hypothesis	Application	Label
Google files for its long awaited IPO.	Google goes public.	IR	entailment
The Republic of Yemen is an Arab, Islamic and independent sovereign state whose integrity is inviolable, and no part of which may be ceded.	The national language of Yemen is Arabic.	QA	entailment

Table 2.1: Examples of sentence pairs from RTE-1 (Dagan et al., 2006)

After the first edition in 2005, new RTE benchmark data sets were proposed every year from 2006 to 2011. While the size of the sets remained approximately the same (a few hundreds of sentence pairs), the process for generating the data was improved and refined at every iteration. Starting from 2008 (RTE-4) (Giampiccolo et al., 2008), the challenge moved from a binary classification setup (entailment/no entailment) to a three class problem (entailment/contradiction/unknown).

Because of the way they were manually built and labelled by human annotators, it is generally accepted that the RTE benchmark data sets are of high quality. However, their small size and lack of training set drastically limit their utility when working on models that need to learn representations on large quantities of data, such as deep neural networks or statistical models. Nevertheless, the data sets can be used as test sets to evaluate DL models after having trained them on larger corpora.

2.2 The SICK Data Set

The *Sentences Involving Compositional Knowledge* (SICK) data set (Marelli et al., 2014) was proposed in 2014 at the SemEval-2014 international workshop on semantic evaluation (Marco Marelli et al., 2014). The objective was to provide a shared benchmark for the evaluation of *Compositional Distributional Semantic Models* (CDSMs) on two specific tasks: semantic relatedness and textual entailment.

SICK consists in 10,000 sentence pairs labelled with a score on a 5-point scale for semantic relatedness and with *entailment/contradiction/neutral* for textual entailment. In the context of SemEval-2014, the data was randomly split in a training and a test set (each containing 50% of the pairs), and it was ensured that labels were evenly distributed between the two halves. Participating systems were evaluated on their ability to correctly predict the scores and labels associated to the examples in the test set.

To construct SICK, the authors extracted 2,000 image captions from the 8K ImageFlickr¹ and SemEval 2012 STS MSRVideo Description data sets and automatically applied a number of transformations to them to obtain sentence pairs. Once this was done, annotators manually labelled the pairs, and cross-evaluation was performed to get gold labels. In the case of textual entailment, a majority vote was used to determine the class associated to each pair. It was measured that, on average, 84% of the participants agreed with the gold labels eventually selected.

The incentive for using image caption data sets as SICK’s basic building block was that they provided sentences describing the same images but with different formulations, which was especially interesting for the task of textual entailment. Captions were also preferred because they often contain very few named entities and many generic terms, which was important to the authors as they wanted to avoid linguistic phenomena that weren’t expected to be accounted for by CDSMs (such as *named entity recognition*) as much as possible.

Table 2.2 presents a few examples of sentence pairs from the SICK data set with their associated gold label for textual entailment (examples were copied from the SemEval-2014 paper).

Premise	Hypothesis	Label
Two teams are competing in a football match	Two groups of people are playing football	entailment
The brown horse is near a red barrel at the rodeo	The brown horse is far from a red barrel at the rodeo	contradiction
A man in a black jacket is doing tricks on a motorbike	A person is riding the bicycle on one wheel	neutral

Table 2.2: Examples of sentence pairs from SICK (Marco Marelli et al., 2014)

While SICK is larger than the RTE data sets by an order of magnitude, its size is still too small to be used in the training of data intensive models such as NNs. In their 2015 paper, Bowman et al. also note that the automatic aspect of the sentence pairs generation introduced *”some spurious patterns into the data”* (S. Bowman et al., 2015). These elements, as well as the absence of other large scale, high quality manually annotated resources, motivated the creation of the *SNLI* corpus. Similarly to the RTE data sets and despite its small size, SICK can be used as a benchmark for deep learning models that were trained on larger data sources.

2.3 The SNLI Corpus

In 2015, Bowman et al. presented the *Stanford Natural Language Inference* (SNLI) corpus (S. Bowman et al., 2015), a large scale, manually generated and annotated data set of sentence pairs labelled for textual entailment. With a total of 570,152 instances, the corpus was the first of its kind, and its impressive size sparked the appearance of numerous deep learning models for natural language inference.

¹<http://nlp.cs.illinois.edu/HockenmaierGroup/8k-pictures.html>

The SNLI corpus is composed of pairs of sentences called *premises* and *hypotheses* and labelled with one of the three classes *entailment*, *contradiction* and *neutral*. The data is divided into a training set containing 550,152 sentence pairs and a development and a test set containing 10,000 pairs each.

To build SNLI, the authors used the Amazon Mechanical Turk² crowd-sourcing platform. There, human workers were presented with series of premises and asked to write three hypotheses for each of them: one that was entailed by the premise (labelled with *entailment*), one that contradicted it (labelled with *contradiction*), and one that wasn't entailed by nor contradicted it (labelled with *neutral*). Specific indications were given to the workers to guide them in their task (advices on sentence length, complexity, etc.), as well as restrictions (it was for example forbidden to reuse the same sentence twice). Examples of sentence pairs and their associated labels are presented in table 2.3 (taken directly from the original paper).

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

Table 2.3: Examples of instances from the SNLI corpus (S. Bowman et al., 2015)

For the premises of SNLI, the authors extracted captions from the Flickr30k corpus (Young et al., 2014), a crowd-sourced data set composed of image descriptions. The motivation for using captions was that it helped solve the problem of *event* and *entity co-reference* in sentence pairs.

Event/entity co-reference refers to the situation where the premise and hypothesis in a pair mention some entity or event, but it cannot be trivially determined whether they are the same or not. In the SNLI paper (S. Bowman et al., 2015), the example of the sentences "A boat sank in the Pacific ocean" and "A boat sank in the Atlantic ocean" is given. In that situation, it is not clear whether it should be considered that the event being referred to is the same or not. If it is, the label associated to the pair should be *contradiction*, as the location of the boat in the hypothesis is different from the one in the premise. However, if the events are considered to be different, the associated label should be *neutral*, because the sentences don't contradict each other since they refer to different boats and accidents.

Using captions helped solve this problem, as all events or entities in the premises belonged to some image on which the hypotheses written by the workers were supposed to be based too. This meant that events and entities mentioned in the hypotheses could always be assumed to be the same as those in the premises when labelling pairs.

Once data collection was complete, an additional validation round was applied on about 10% of the corpus. Instances from the data set were presented to the

²<https://www.mturk.com/>

workers without their associated class, and they were asked to label them. Four different persons validated each pair, and a majority vote was used to determine the gold label. Sentence pairs for which no agreement was reached were kept in the corpus but labelled with ”-”, to indicate that no gold label could be selected. The rate of agreement during this phase was of 98%, with unanimity obtained in about 58% of cases.

Aside from the corpus described above, Bowman et al. also proposed several models trained and tested on the data in their paper. The two best performing ones were a lexicalised classifier and a simple *Long Short Term Memory* (LSTM) neural network. These were defined as the baselines to beat when the paper was published. Additionally, the LSTM was tested on the SICK corpus with transfer learning. The authors first trained the model on SNLI and then fine-tuned it on SICK’s training set. With this approach, they obtained new state-of-the-art results, which showed the potential of the SNLI data set for training efficient deep learning models on the task of recognising entailment.

Similarly to the RTE data sets, the manual construction and annotation of the SNLI corpus by human workers make it a high quality resource. Its large size also allows it to be particularly appropriate for uses in modern deep learning approaches to NLI.

However, the corpus has its limitations. Williams et al. (2018) explain in their MultiNLI paper that because all sentences in SNLI were build on a single type of textual resource (namely image captions), they do not allow for good generalisation on other kinds of texts and lack certain important linguistic phenomena (such as temporal reasoning or modality, among other examples). These were some of the reasons for the creation of the MultiNLI corpus.

2.4 The MultiNLI Corpus

The *Multi-Genre Natural Language Inference* (MultiNLI) corpus (Williams et al., 2018) was created at the University of New York in 2017 to remedy the shortfalls of SNLI. It consists in 432,702 pairs of sentences labelled for textual entailment and covers a wide range of textual styles and topics. The data is split in a training and development set that are available online³, as well as test sets that can only be accessed through Kaggle competitions in unlabelled form⁴⁵.

MultiNLI is very similar to SNLI both in its structure and in the way it was constructed: the sentence pairs that compose it were produced through a crowd-sourcing effort that followed the same protocol, and they have the same form.

The main difference between the two corpora is the type of textual resources that were used to produce their premises. Compared to SNLI, many more types of text were used for MultiNLI. More specifically, the authors extracted premises from ten different types of sources written in English. Nine of them were part of the *Open*

³<https://www.nyu.edu/projects/bowman/multinli/>

⁴<https://www.kaggle.com/c/multinli-matched-open-evaluation>

⁵<https://www.kaggle.com/c/multinli-mismatched-open-evaluation>

American National Corpus (OANC), which contains transcriptions of real world conversations, reports, speeches, letters, non-fiction works, articles from magazines, travel guides and short posts on linguistics for non-specialists. The tenth genre of text used in MultiNLI was fiction and contained a compilation of open access works written between 1912 and 2010, covering different styles such as science-fiction, mystery or adventure.

One of the problems that Williams et al. identified in SNLI was that it was *“not sufficiently demanding to serve as an effective benchmark for NLU”* (Williams et al., 2018). Hence, in order to make MultiNLI more difficult, they split its data such that only five genres of text were covered in its training set, and its testing set was divided in two categories: a *matched* set only containing premises extracted from the same genres as the training set, and a more challenging *mismatched* set containing premises from all ten genres selected during data collection. The insight was that the matched set would allow to *“explicitly evaluate models [...] on the quality of their sentence representations within the training domain”*, whereas the mismatched version would allow to test *“their ability to derive reasonable representations in unfamiliar domains”* (Williams et al., 2018).

Table 2.4 below presents some examples of sentence pairs extracted from the MultiNLI corpus (taken from the data set’s official website⁶).

Text type	Premise	Hypothesis	Label
Letters	Your gift is appreciated by each and every student who will benefit from your generosity.	Hundreds of students will benefit from your generosity.	neutral
Telephone	yes now you know if if everybody like in August when everybody’s on vacation or something we can dress a little more casual or	August is a black out month for vacations in the company.	contradiction
9/11 report	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.	entailment

Table 2.4: Examples of instances from the MultiNLI corpus (Williams et al., 2018)

To verify that their corpus did indeed improve the difficulty compared to SNLI, the authors of MultiNLI trained and tested several baselines on it, as well as the then state of the art ESIM model (Chen, Zhu, Ling, Wei, et al., 2017a). They wrote their own implementation of ESIM and tested it on SNLI, which yielded 86.7% accuracy (meaning that the model correctly classified 86.7% of the instances it saw in SNLI’s test set). On MultiNLI, their implementation only performed at 72.4% accuracy on the matched set and 71.9% on the mismatched version, effectively proving that their new corpus represented a greater challenge than SNLI for natural language inference.

⁶<https://www.nyu.edu/projects/bowman/multinli/>

While MultiNLI is now widely accepted as the de facto standard for training and evaluating NLI models, the corpus is not devoid of flaws. In particular, two papers published in 2018 (Gururangan et al., 2018; Poliak et al., 2018) showed that it was possible to predict the labels associated to sentence pairs in SNLI and MultiNLI with relatively high accuracy by solely looking at the hypotheses. The explanation given by the two works to explain this phenomenon was that the corpora suffered from annotation artifacts, such as specific sentence length or choices of words by the annotators for given classes.

2.5 The Breaking NLI Data Set

In 2018, Glockner et al. proposed a new benchmark to evaluate whether the models trained to perform NLI were efficient at solving problems that involve lexical inferences and world knowledge. The corpus, named the *Breaking NLI* data set (Glockner et al., 2018), consists in only a test set of 8,193 sentence pairs and is meant to be used to evaluate models previously trained on SNLI.

To build their data set, Glockner et al. extracted premises from SNLI’s training set and applied automatic transformations on them to produce hypotheses where only a single word has been replaced. Replacement words were chosen to generate hypotheses that were either entailed by the selected premises or contradicted them (though neutral examples were also obtained in some cases as a by-product). After the sentence pairs and their associated label were obtained with the automatic procedure, they were further validated by human annotators through a crowd-sourced effort, to ensure their correctness. Examples of pairs and their labels are provided in table 2.5 (directly taken from the paper).

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 2.5: Examples of instances from the *Breaking NLI* data set (Glockner et al., 2018)

Once they had completed data collection and validation, the authors evaluated a number of models that performed well on SNLI and MultiNLI on their own test set. All models performed significantly worse on their data, except for one: KIM (Chen, Zhu, Ling, Inkpen, et al., 2018a), which makes use of external lexical information to perform classification.

The results show that most models trained on the current best corpora for NLI have poorer generalisation capability than was previously thought, and that further improvements in the NLI task definition would be necessary to alleviate this problem.

2.6 Other resources

In addition to the data sets mentioned in this chapter, other resources exist for the task of recognising textual entailment. These won't be discussed in depth here, as we consider them of somewhat lesser relevance in the specific context of deep learning for NLI. Nevertheless, we list them below for the sake of completeness:

- The *denotation graph* (Young et al., 2014) consists in a large hierarchical set of sentences connected to each other through the relation of entailment. As with the SICK data set, Bowman et al. explain in their SNLI paper (S. Bowman et al., 2015) that the automatic generation of the denotation graph makes it too noisy to be usable in the training of data intensive models.
- The *Cross-lingual Natural Language Inference* (XNLI) corpus (Conneau, Rinott, et al., 2018) is a data set that extends the development and test sets of the MultiNLI corpus with 7,500 human-annotated pairs of sentences in 15 different languages.

While the pairs follow the exact same structure as those in the SNLI and MultiNLI data sets, the primary focus of the XNLI corpus is not on recognising textual entailment, but rather on providing a strong benchmark for the evaluation of systems on the task of *cross-lingual natural language understanding*.

Chapter 3

Deep Learning Models for NLI

Since the release of the SNLI corpus in 2015, a wide array of deep learning models have been devised for the task of natural language inference. Those models can roughly be regrouped into three main categories: sentence vector-based models, sentence matching models and transfer learning approaches. These are explored into more detail in separate sections of this chapter.

3.1 Sentence Vector-Based Models

As mentioned in the MultiNLI paper (Williams et al., 2018), the wide variety of linguistic phenomena covered by natural language inference not only makes it a good benchmark for NLU, but also an excellent supervised task for the learning of *sentence embeddings* (vector representations that capture the semantics of sentences). For this reason, there exist a large number of models focused on learning general sentence representations from NLI in the literature. These models are often referred to as *sentence vector-based*.

The general architecture of sentence vector-based models consists in two main components: a *sentence encoder* (or *sentence model*) and a *classifier* (or *matching layer*). The task of the sentence encoder is to learn generic representations for the premises and hypotheses in some NLI problem, and the classifier then has to somehow combine these representations to predict the relationship that exists between the two sentences. This structure is often referred to as the *Siamese architecture* (Bromley et al., 1994), represented in figure 3.1. Note that the two sentence encoders in the image share the same weights (the same network is applied both to the premise and hypothesis).

All sentence vector-based models presented in this section use the same type of classifier to perform predictions on NLI. The main difference between them thus lies in the way sentence representations are learned by the encoder. Approaches to sentence encoding can be subdivided into three principal categories: *sequential*, *tree-based* and *self attention-based*.

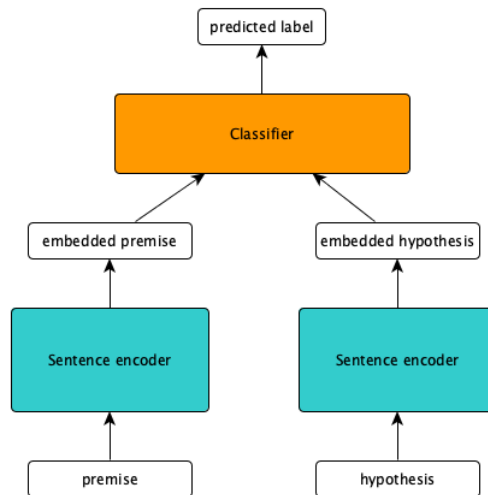


Figure 3.1: Representation of the Siamese architecture for NLI

3.1.1 Classification Layer

As mentioned above, the classification layer used by all the sentence vector-based models presented in this document is virtually the same (with only hyper-parameters such as layers size varying). The architecture, first introduced in a paper by Mou et al. (2015), is illustrated in figure 3.2.

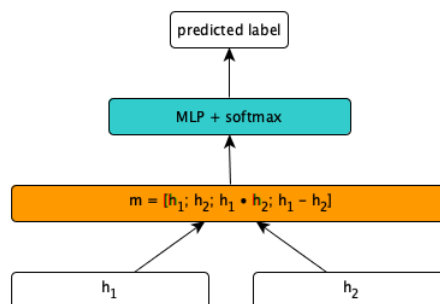


Figure 3.2: Classifier architecture in sentence vector-based models

In the figure, h_1 and h_2 are vector representations learned by a model’s sentence encoder for the premise and hypothesis in some NLI task. These vector representations, as well as the element-wise product and difference between them, are concatenated into a single vector m , which is then passed through a *Multi-Layer Perceptron* (MLP). To associate a probability to each possible class in the NLI task, a *softmax* function is applied on the output of the MLP.

3.1.2 Sequential Encoders

Sequential sentence encoders are models that use *Recurrent Neural Networks* (RNNs) to learn representations for sentences. RNNs are a kind of neural network specifically designed to process sequential data in time steps. They read elements in sequences (such as words in sentences) one by one and keep some form of memory

between steps. This allows them to capture information about the relationships that exist between elements in a sequence, which makes them particularly appropriate for working on natural language sentences. Different types of RNNs include plain RNNs, *Gated Recurrent Unit* (GRU) and *Long Short Term Memory* (LSTM) networks. A good introduction on RNNs, and more specifically LSTMs, can be found on Christopher Olah’s blog¹.

In the paper *“Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”*, Conneau, Kiela, et al. (2017) investigate several architectures involving RNNs for sentence encoding. In particular, they propose in a first approach to encode sentences by passing them through a unidirectional, single-layer GRU or LSTM network and taking the final state of the RNN as representation. In a second proposition, a *bidirectional LSTM* (bi-LSTM) is used, and *max* or *average pooling* is applied over each dimension of the network’s hidden states to extract a fixed-length vector for a sentence.

Subsequent works by Nie and Bansal (2017) and Talman et al. (2018) propose models using *stacked bi-LSTMs* with shortcut connections and max pooling over the output to encode sentences. The general architecture of the models is illustrated in figure 3.3. There are only small differences between the propositions in the two papers. Nie and Bansal pass both the initial word embeddings and the output of each stacked layer to the next layers with shortcut connections, and they apply max pooling only on the result of the last bi-LSTM. In the model proposed by Talman et al., only the word embeddings are shortcut, but max pooling is applied on the output of each stacked layer, and the results are concatenated into a single sentence representation.

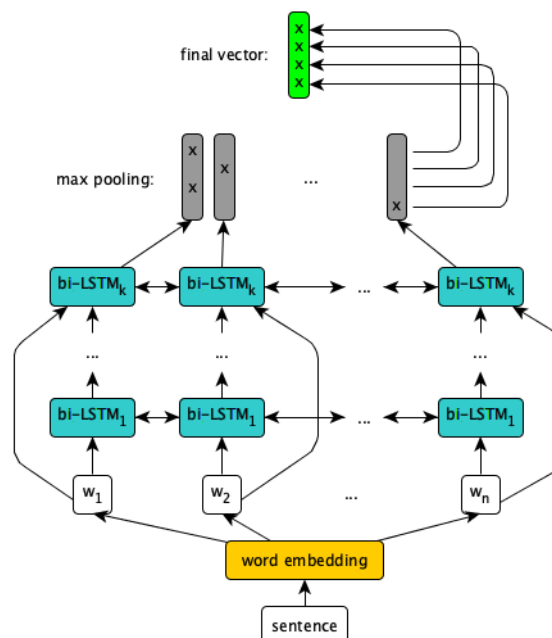


Figure 3.3: k -layers stacked bi-LSTMs with shortcut connections for sentence encoding

¹<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Table 3.1 summarises the reported accuracies of the models presented in this section on SNLI and MultiNLI’s test sets.

Model	SNLI	MultiNLI-m	MultiNLI-mm
LSTM (Conneau, Kiela, et al., 2017)	80.7	-	-
GRU (Conneau, Kiela, et al., 2017)	81.8	-	-
Bi-LSTM avg. (Conneau, Kiela, et al., 2017)	78.2	-	-
Bi-LSTM max. (Conneau, Kiela, et al., 2017)	84.5	-	-
Shortcut-stacked encoder (Nie and Bansal, 2017)	86.1	74.6	73.6
HBMP (Talman et al., 2018)	86.6	73.7	73.0

Table 3.1: Reported accuracy (%) of sequential sentence encoders on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets

3.1.3 Tree-based Encoders

Tree-based encoders make use of the parse structure of sentences to learn their representations. This means that syntactic information is explicitly taken into account by the model to produce sentence embeddings.

The *Tree-based Convolutional Neural Network* (TBCNN), presented by Mou et al. (2015), is one of the first tree-based sentence encoders ever proposed for RTE. The general idea of the model is to apply a *Convolutional Neural Network* (CNN) over the dependency parse tree of a sentence to learn a representation for it. More specifically, Mou et al. propose in their paper to slide a set of *two-layers sub-tree feature detectors* over the parse tree of a sentence to learn *feature maps* for each word in it. The feature detectors are convolution filters applied on every sub-tree in the dependency parse tree of a sentence. They are each specialised to capture information about a specific grammatical relation between words. Once the detectors have been applied, the resulting feature maps are combined together by applying a max pooling operation over each of their dimensions, and then passing the result through some *feed-forward* neural network. This produces a fixed-length vector representation for a sentence.

Aside from CNNs, other types of neural networks can be applied on the parse structure of sentences to encode them. This is exactly what is proposed in two works by S. R. Bowman et al. (2016) and Choi et al. (2017), which use *Tree-structured Recursive Neural Networks* (TreeRNNs) to learn sentence representations.

TreeRNNs are a special type of RNN that work on binary parse trees and propagate information upstream along them. This particular approach is interesting for sentence encoding because it captures both syntactic and semantic information over a whole sequence’s tree from the bottom up. This makes it possible to take the output of the network at the top of a sentence’s tree and use it as an efficient representation for it.

S. R. Bowman et al. (2016) introduce the *Stack-augmented Parser-Interpreter Neural Network* (SPINN). Inspired by *shift-reduce parsing*, SPINN is capable of both building the binary parse tree of a sentence and processing it in a single left-to-right pass over its tokens.

The same is true of the model proposed by Choi et al. (2017), which introduces a new kind of TreeRNN called *Gumbel Tree-LSTM* to learn the parse structure of a sentence as it is being read.

This capability of the two models to parse sentences as they learn representations for them makes them particularly fast, because they only need one pass over a sentence to encode it, and they can work on batches of sentences instead of single sequences at a time (which is a big limitation of other TreeRNNs).

Table 3.2 summarises the reported classification accuracies of the tree-based sentence encoders presented in this section on SNLI’s test set. No results are available on MultiNLI for those models.

Model	Accuracy (%)
TBCNN (Mou et al., 2015)	82.1
SPINN (S. R. Bowman et al., 2016)	83.2
Gumbel Tree-LSTM (Choi et al., 2017)	86.0

Table 3.2: Reported accuracy of tree-based sentence encoders on SNLI’s test set

3.1.4 Self Attention-based Encoders

As explained by Shen, Zhou, Long, Jiang, S. Pan, et al. (2017), *attention* is a special mechanism used to compute an alignment score between the elements of a sequence and some *query*. In particular, given a vector q representing a query and a sequence of vectors $x = [x_1, x_2, \dots, x_n]$, an attention function $a(x_i, q)$ computes the degree of *similarity* or *dependency* between each $x_i \in x$ and q . A softmax function is then applied on the resulting scores to produce a probability distribution describing how likely each element x_i is to contribute to the information in the query q .

Typically, the attention scores computed for a sequence x are used in some weighted operation involving the $x_i \in x$ (such as a sum) to build a summary of the relationship between x and q . In the specific case of sentence encoding, this idea is applied on the words of a sentence to learn a general representation for it (which can be seen as a summary of the sentence’s meaning). The name *self attention* (or *inner attention*) comes from the fact that attention is computed on the same sentence that is being encoded, as opposed to other situations in NLP where the mechanism is used on pairs of sentences to extract the dependencies between their elements (more on this in section 3.2).

There are multiple ways of using self attention to encode sentences’ meanings in vector representations. Y. Liu et al. (2016) were among the first to do it in the context of NLI. In their paper, they first pass the n word embeddings e_i of a sentence s through a bi-LSTM, which produces hidden states $h_i, i \in [1, \dots, n]$. They then apply average pooling on the h_i and use a feedforward network to compute the attention between the resulting vector and the hidden states of the bi-LSTM. This produces weights α_i that are used in a weighted sum of the h_i to get a vector representation m for the sentence s . The architecture is illustrated in figure 3.4. If we map Liu et al.’s attention mechanism to the one described earlier in this section, we see that the result of average pooling corresponds to the query vector q ,

the bi-LSTM’s hidden states to the x_i , and the feed-forward network to the attention function $a(x_i, q)$.

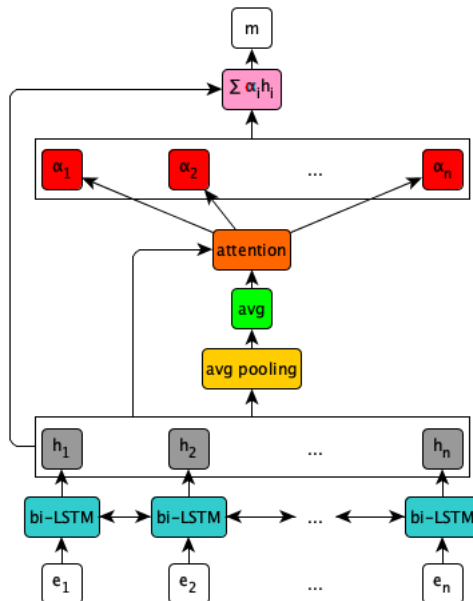


Figure 3.4: Inner attention mechanism (Y. Liu et al., 2016)

Lin et al. (2017) and Conneau, Kiela, et al. (2017) are inspired by the attention mechanism described by Y. Liu et al. (2016), but they propose to use multiple attentions computed between the bi-LSTM’s hidden states and learned query vectors, instead of a single attention with the result of average pooling as query. To force the attentions to focus on different parts of a sentence, a special penalisation term is used.

Shen, Zhou, Long, Jiang, S. Pan, et al. (2017) further develop the idea in their *Directional Self Attention Network* (DiSAN), which introduces the concept of *multi-dimensional self attention*, a special form of attention where weights are computed for each dimension of a vector (producing weight vectors instead of scalar attention scores). In a later publication, Shen, Zhou, Long, Jiang, Sen Wang, et al. (2018) present the *Reinforced Self Attention Network* (ReSAN), an improved DiSAN model that combines hard and soft attention mechanisms to learn representations of sentences. Soft attention refers to the situation where a probability distribution over the elements of a sequence is produced (i.e. continuous attention weights), whereas hard attention computes binary attention weights (selecting a subset of the elements in a sequence).

Chen, Zhu, Ling, Wei, et al. (2017b) develop a model based on *gated attention* (a form of attention using the bi-LSTM’s gates in its computation), and Chen, Ling, et al. (2018) use several forms of multi-dimensional self attention with *generalised pooling*. In other work, Munkhdalai and Yu (2017) propose a new neural architecture based on memory and self attention for sentence encoding, and Yoon et al. (2018) apply self attention on top of a CNN in their *Dynamic Self Attention* (DSA) network to learn representations. Finally, Im and Cho (2017) use the *Transformer* architecture (Vaswani et al., 2017) and a form of attention sensitive to the distance between words in their *Distance-based Self Attention Network*.

Table 3.3 summarises the reported accuracies of self attention-based models on the SNLI and MultiNLI test sets.

Model	SNLI	MultiNLI-m	MultiNLI-mm
Inner attention (Y. Liu et al., 2016)	83.3	-	-
Neural Semantic Encoder (Munkhdalai and Yu, 2017)	84.6	-	-
Structured Self Attentive Network (Lin et al., 2017)	84.4	-	-
Inner attention (Conneau, Kiela, et al., 2017)	82.5	-	-
Gated attention bi-LSTM (Chen, Zhu, Ling, Wei, et al., 2017b)	85.5	72.8	73.6
DiSAN (Shen, Zhou, Long, Jiang, S. Pan, et al., 2017)	85.6	71.0	71.4
Distance-based Self Attention Network (Im and Cho, 2017)	86.3	74.1	72.9
ReSAN (Shen, Zhou, Long, Jiang, Sen Wang, et al., 2018)	86.3	-	-
bi-LSTM generalised pooling (Chen, Ling, et al., 2018)	86.6	73.8	74.0
DSA (Yoon et al., 2018)	87.4	-	-

Table 3.3: Reported accuracy (%) of self attention-based sentence encoders on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets

3.2 Sentence Matching Models

Natural language inference is a task that involves comparing pairs of sentences to predict the relation linking them. This means that interactions between premises and hypotheses play an essential role in the decision whether they entail each other or not. However, because the goal of sentence vector-based models is to learn representations for single sentences that can be used in downstream tasks, they do not capture any information about interactions between sentence pairs to recognise entailment.

Sentence matching models, on the contrary, do exactly that. While this makes them less applicable in transfer learning, it gives them a clear advantage on recognising textual entailment, as is reflected by their classification accuracy on NLI tasks.

There are multiple ways of modelling interactions between sentences, but almost all of them involve attention mechanisms similar to the one described in section 3.1.4.

Rocktäschel et al. were probably the first to propose a deep sentence matching model for NLI. The system they describe in their paper *Reasoning about Entailment with Neural Attention* (Rocktäschel et al., 2015) uses two LSTMs with word-by-word attention to learn a representation of the interactions between a premise and a hypothesis.

The architecture of the model is illustrated in figure 3.5. In the image, the $e_i^p, i \in [1, \dots, l]$ are word embeddings for the premise, $e_j^h, j \in [1, \dots, m]$ word embeddings for the hypothesis, $LSTM^p$ and $LSTM^h$ two LSTMs to encode the premise and hypothesis respectively, h_i^p the hidden states of $LSTM^p$ and h_j^h those of $LSTM^h$. The authors use a feed-forward network to compute attention between each h_j^h (the encoded words of the hypothesis) and all the h_i^p (the encoded words of the premise) to produce attention weights α_i^j . For each h_j^h , an attention vector r_j is then obtained by computing a weighted sum a_j of the attended h_i^p with the α_i^j ($a_j = \sum_{i=1}^l \alpha_i^j h_i^p$) and merging it with r_{j-1} , the attention vector computed for the previous word in the hypothesis ($r_j = a_j + \tanh(W^r r_{j-1})$, W^r is a learnable parameter). Finally, the last attention vector r_m is taken as a representation of all the interactions between

the premise and hypothesis and merged with the last hidden state h_m^h of $LSTM^h$. The result is passed through a classification layer with softmax activation to predict the label associated to the pair.

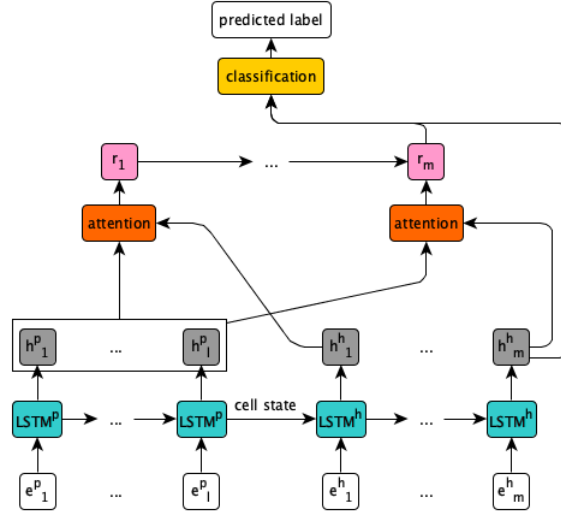


Figure 3.5: Word-by-word attention mechanism (Rocktäschel et al., 2015)

In subsequent work, Shuohang Wang and Jiang (2016), Sha et al. (2016) and P. Liu et al. (2016) are inspired by the proposition of Rockstäschel et al. to build their own LSTMs with word-by-word attention for NLI.

Shuohang Wang and Jiang (2016) reuse Rockstäschel et al’s architecture, but they pass each h_j^h and a_j through an additional layer they call *match-LSTM* (mLSTM) to merge their representations. They then use the last hidden state of the mLSTM for classification.

Sha et al. (2016) first encode the premise with a regular LSTM network and concatenate its hidden states in a matrix P . Then, they pass P and the word embeddings of the hypothesis through a special LSTM called *re-read LSTM* (rLSTM), which combines the encoded words of the hypothesis with a weighted sum of the vectors in P . The weights of the sum are computed with attention. They apply average pooling on the outputs of the rLSTM and use the result for classification.

P. Liu et al. (2016) propose something slightly different with their *coupled-LSTMs* (c-LSTMs), two inter-dependent LSTMs that encode the premise and hypothesis in a pair by using both their own previous hidden states and the ones from the other LSTM at different time steps to produce outputs. The authors stack multiple coupled-LSTM on top of each other to get their best performance on NLI with this approach.

In the paper “*A Decomposable Attention Model for Natural Language Inference*”, Parikh et al. (2016) propose to apply neural attention directly on the word embeddings of a sentence pair to perform classification. First, they use a feed-forward network to compute attention scores between each word embedding in the premise, denoted $e_i^p, i \in [1, \dots, l]$, and those in the hypothesis, denoted $e_j^h, j \in [1, \dots, m]$. A softmax is applied on the result, which produces attention weights α_{ij} that are used to compute, for each word e_i^p in the premise, a weighted sum of the words in the hypothesis $a_i = \sum_{j=1}^m \alpha_{ij} e_j^h$, and the inverse for each word e_j^h in the hypothesis, $b_j = \sum_{i=1}^l \alpha_{ij} e_i^p$. Then, each pair (e_i^p, a_i) and (e_j^h, b_j) is concatenated and passed

through a feed-forward network G to produce comparison vectors $v_i^p = G([e_i^p; a_i])$ and $v_j^h = G([e_j^h; b_j])$. Finally, the v_i^p and v_j^h are aggregated by summing them, $v^p = \sum_{i=1}^l v_i^p$, $v^h = \sum_{j=1}^m v_j^h$, and the results are concatenated and passed through a final feed-forward network for classification, $\hat{y} = F([v^p; v^h])$. The complete architecture is called *attend-compare-aggregate* and is illustrated in figure 3.6.

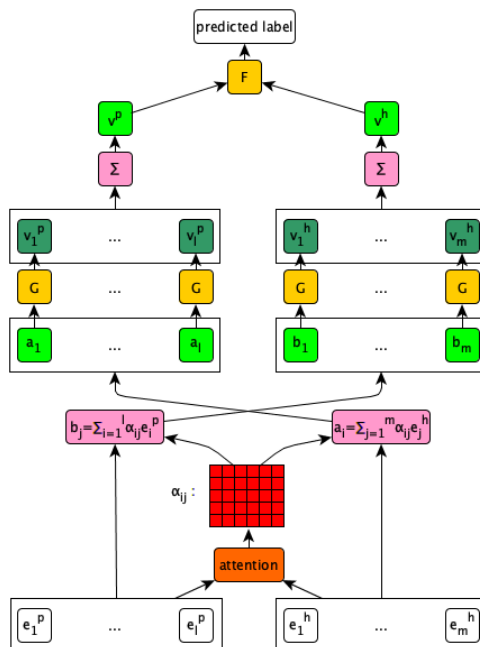


Figure 3.6: Attend-compare-aggregate architecture (Parikh et al., 2016)

The idea of composing attention both from the premise to the hypothesis and from the hypothesis to the premise was reused in numerous models after the publication by Parikh et al. in 2016. A famous example is the *Enhanced Sequential Inference Model* (ESIM) (Chen, Zhu, Ling, Wei, et al., 2017a), depicted in figure 3.7. The model first encodes the word embeddings of the premise and hypothesis with a bi-LSTM, and it computes attention between the outputs in the same way as Parikh et al. to produce "attention vectors" a_i and b_j . Then, each encoded word h_i^p of the premise is concatenated with its corresponding a_i , as well as with the element-wise difference and product with it, giving a vector $u_i^p = [h_i^p; a_i; h_i^p - a_i; h_i^p \cdot a_i]$. The same is done with the h_j^h and b_j : $u_j^h = [h_j^h; b_j; h_j^h - b_j; h_j^h \cdot b_j]$. The u_i^p and u_j^h are passed through a feed-forward layer to reduce their dimensionality (not shown in the figure for readability), and then through a second bi-LSTM, outputting hidden states v_i^p and v_j^h for the premise and hypothesis, respectively. Finally, the vectors are aggregated through average and max pooling, and the results concatenated and passed through a MLP with a softmax on the output for classification. Chen et al. also propose in their paper to use a tree-LSTM along with ESIM to take syntactic information into account when predicting inference, which slightly improves the model's performance.

Z. Wang et al. (2017) present the *Bilateral Multi-Perspective Matching* (BiMPM) network, a model essentially following the same architecture as ESIM, with differences only in the way attention is computed and composed.

Tay et al. follow with another model called *Comprop Alignment-Factorised Encoders*

(CAFE) (Tay et al., 2018), which has a similar architecture as ESIM but composes attention information differently, with an *alignment factorisation layer*.

The *Dependant Reading Bidirectional LSTM* (DR-BiLSTM) model from Ghaeini et al. (2018) also works the same way as ESIM, with the exception that it uses special inter-dependant bi-LSTMs for the encoding of the premise and hypothesis instead of regular ones.

Finally, Kim et al.’s *Densely-connected Recurrent and Co-attentive neural Network* (DRCN) (Kim et al., 2018) adopts an architecture similar to ESIM’s, but it stacks multiple RNNs and attention layers on top of each other. Auto-ecoders are used between stacked layers to reduce the representation’s dimensionality (since it grows with the number of layers). Figure 3.8, taken from the original paper, illustrates the concepts of the model. The top-right part of the image shows the specific number and disposition of layers used by the authors in the publication.

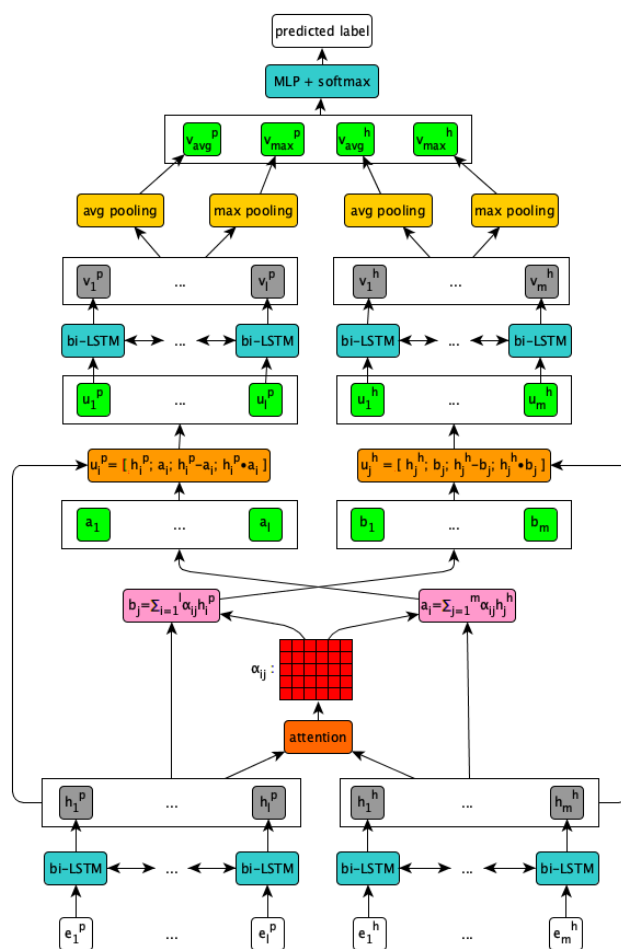


Figure 3.7: ESIM architecture (Chen, Zhu, Ling, Wei, et al., 2017a)

In the *Densely Interactive Inference Network* (DIIN) (Gong et al., 2018), the authors use a *highway network* followed by a self attention mechanism to encode the premise and hypothesis of a sentence pair. Highway networks (Srivastava et al., 2015) are a special kind of gated feed-forward networks specifically designed to be deeper than regular multi-layer perceptrons. After the encoding step, a form of multi-dimensional attention is computed between the obtained representations, producing a 3-dimensional attention matrix. A convolutional feature extractor is

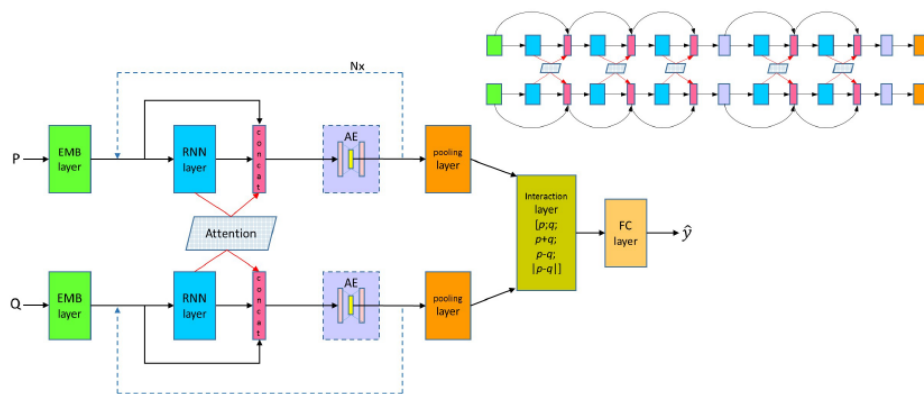


Figure 3.8: DRCN architecture (Kim et al., 2018)

then applied on it to retrieve information about interactions between the sentences, and the result is flattened and passed through a MLP with softmax to predict the class associated to the pair of inputs.

Only very few papers investigate the use of external knowledge for NLI. All of them propose to use sentence matching models with the inclusion of some outside information to improve their performance on recognising entailment.

Chen, Zhu, Ling, Inkpen, et al. (2018b) introduce the *Knowledge-based Inference Model* (KIM), a model that follows the general architecture of ESIM, but where information about the lexical relations between words in the premise and hypothesis is additionally used to improve the model’s performance. In particular, lexical relations between words such as synonymy, hypernymy or antonymy are extracted from *Wordnet* (Miller, 1995) and used in the attention layer.

In another paper, Zhang et al. (2018) propose to use *Semantic Role Labelling* (SRL), a task where the objective is to predict the predicate-argument relations in a sentence, to improve an existing model on NLI. The authors apply SRL on premises and hypotheses to learn the semantic roles for the words they contain, and the results are used as additional information in the ESIM model, which improves its performance.

Table 3.4 summarises the reported accuracies of all the sentence matching models presented in this section on SNLI and MultiNLI’s test sets.

3.3 Transfer Learning Approaches

In *transfer learning*, models that were first trained on some task are reused and fine-tuned to perform other tasks. The approach is often used in situations where resources are very scarce for a given problem, but there are large amounts of training data available for some other, related task. In those cases, a model is first trained on the objective where lots of data are available, and its parameters are then fine-tuned for the problem with fewer resources.

Over the past years, there have been numerous examples of successful applications of transfer learning to deep learning models. While earlier examples have mostly been in computer vision (thanks to the release of the huge *ImageNet* data

Model	SNLI	MultiNLI-m	MultiNLI-mm
Word-by-word attention (Rocktäschel et al., 2015)	83.5	-	-
Match-LSTM (Shuohang Wang and Jiang, 2016)	86.1	-	-
rLSTM (Sha et al., 2016)	87.5	-	-
Stacked TC-LSTMs (P. Liu et al., 2016))	85.1	-	-
Attend-Compare-Aggregate (Parikh et al., 2016)	86.3	-	-
ESIM (Chen, Zhu, Ling, Wei, et al., 2017a)	88.0	76.8	75.8
ESIM + Tree-LSTM (Chen, Zhu, Ling, Wei, et al., 2017a)	88.6	-	-
BiMPM (Z. Wang et al., 2017)	87.5	-	-
CAFE (Tay et al., 2018)	88.5	78.7	77.9
DR-BiLSTM (Ghaeini et al., 2018)	88.9	-	-
DRCN (Kim et al., 2018)	88.9	80.6	79.5
DIIN (Gong et al., 2018)	88.0	80.0	78.7
KIM (Chen, Zhu, Ling, Inkpen, et al., 2018b)	88.6	77.2	76.4
SRL (Zhang et al., 2018)	89.1	-	-

Table 3.4: Reported accuracy (%) of sentence matching models on SNLI and MultiNLI’s matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets

set (Russakovsky et al., 2015)), more recent applications to NLP tasks have also shown to provide impressive performance gains.

In NLP, almost unlimited resources are available in the form of unlabelled, free text. This data can be used in the unsupervised training of deep learning models to capture information about the general form and structure of language. Models trained in this manner can then be later fine-tuned for more specific tasks requiring human annotated resources, which are much more difficult to produce and hence much scarcer.

Transfer learning can also be used between supervised tasks that share similarities or are somehow complementary.

In the specific case of NLI, although reasonably large quantities of data are available for training, the application of transfer learning has allowed models to reach significantly higher classification accuracy than the previous state-of-the-art on famous data sets like SNLI or MultiNLI. These results show that pre-training models on unsupervised tasks to allow them to better model language seems to also make them more efficient on natural language understanding.

In the cases where transfer learning is used between natural language inference and other supervised tasks, performance improvements show that information about linguistic phenomena captured in other tasks and overlooked by NLI can help models recognise entailment with more accuracy.

3.3.1 Transfer Learning from Supervised Tasks

McCann et al. (2017) propose to apply transfer learning between *Machine Translation* (MT) and other NLP problems by learning special *Context Vectors* (CoVe) that can be reused in downstream tasks. The general idea of the model, illustrated in figure 3.9 (taken directly from McCann et al.’s paper), is to first train the sentence encoder of a MT task to learn context sensitive representations for words so they can be translated accurately, and then reuse it in other downstream tasks to encode input words.

The insight behind this approach is that, in many NLP problems, models need representations for words that are specific to the contexts in which they appear to perform well. ”Traditional” word embedding methods, however, only produce single vectors for words that are context-independent.

In machine translation, it is only after word embeddings are passed through a model’s encoder that their representations become context specific. Because MT is a problem where large quantities of parallel data are available for training, it also happens to be well adapted for transfer learning. This justified the use of a pre-trained MT encoder to contextualise words in downstream tasks.

For the particular problem of natural language inference, McCann et al. show in their paper that using CoVe improves the accuracy of the model they train on the SNLI corpus.

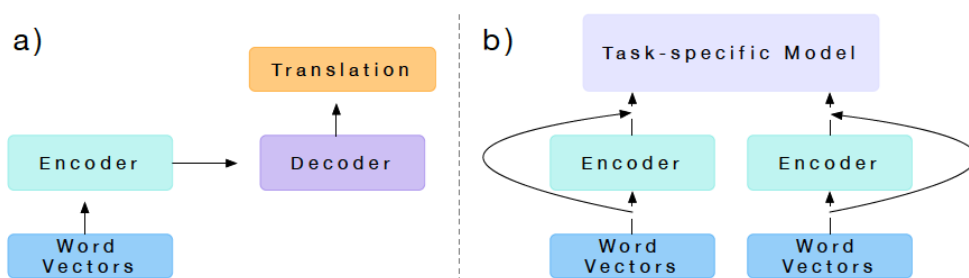


Figure 3.9: Transfer learning in CoVe (McCann et al., 2017)

With the *Discourse Marker Augmented Network* (DMAN), B. Pan et al. (2018) propose to apply transfer learning between the supervised task of *Discourse Marker Prediction* (DMP) and NLI.

In DMP, the objective is to predict the *discourse marker* which connects the two halves S_1 and S_2 of a sentence S . Discourse markers are words that carry information about the relation between parts of a sentence, such as ”and”, ”or” or ”but”. The authors of DMAN underline in their paper how these words *intuitively correspond to the intent of NLI, such as ’but’ to contradiction, ’so’ to entailment, etc.*” (B. Pan et al., 2018), which is why they choose to transfer knowledge from DMP to RTE. To do so, they first train a sentence encoder on a DMP task, and then integrate it in a model specifically designed to recognise entailment. The sentence encoder’s parameters are fine-tuned during training on the NLI task. With this approach, new state-of-the-art results were obtained at the time of publication.

3.3.2 Transfer Learning from Unsupervised Tasks

Language modelling is a common task in NLP where systems called *Language Models* (LMs) are trained to learn about the structure of language in an unsupervised manner. Usually, the objective for LMs is to predict the probability of the next word in a sentence given the previous ones. Formally, if $s = [w_1, w_2, \dots, w_n]$ is a sentence of n words w_i , the goal of a language model is to predict $P(w_i | w_1, \dots, w_{i-1}), \forall i \in [1, \dots, n]$. Modern LMs learn to predict such probabilities statistically on large corpora of unlabelled data, and the current state-of-the-art is obtained with deep neural networks. It is generally accepted that, with sufficient amounts of data available for training,

LMs are able to learn good representations for language. Since unsupervised text exists in almost unlimited quantities, this makes language modelling particularly appropriate for transfer learning.

Peters et al. (2018) propose in their paper "Deep Contextualised Word Representations" to use transfer learning to contextualise word vectors in NLP tasks.

Their approach to transfer knowledge from one task to the other is similar to McCann et al.'s with CoVe: an encoder is first trained on some particular task to contextualise word embeddings, and it is then integrated in other models to encode their input words. However, the task used by Peters et al. for pre-training is very different from the one in CoVe.

In their *Embeddings from Language Models* (ELMo), Peters et al. first train a deep *bidirectional Language Model* (biLM) on some unsupervised language modelling task and then use a linear combination of the biLM's internal states to represent input words in downstream tasks.

ELMo's authors show that using a combination of the biLM's internal states produces richer word representations than simply taking the network's output, for example, because different layers of a biLM capture different levels of information about language (lower layers are often more focused on structure and syntax, while layers at the top usually learn about meaning).

When they integrate ELMo in existing models for various NLP tasks, and in particular the well-known ESIM for NLI, Peters et al. report increases in classification accuracy, justifying their approach.

In the paper "Improving Language Understanding by Generative Pre-Training" (Radford et al., 2018), the authors present the *Generative Pre-training Transformer* (GPT), a language model based on the *transformer* architecture (Vaswani et al., 2017).

In order to apply transfer learning, the GPT is first trained on the language modelling objective presented in the first paragraph of this section with unlabelled data, and it is later fine-tuned for various natural language understanding tasks. Figure 3.10 (taken from the GPT paper) illustrates a high-level view of the model's architecture on the left, as well as a representation of the way the inputs for different tasks are modified to fine-tune GPT on the right. In the particular case of entailment, the premise and hypothesis of a NLI task are concatenated into a single sequence (separated by special delimiters), the result is passed through the transformer language model, and the model's final output is used as input in a linear classification layer. With this approach, Radford et al. manage to obtain new state-of-the-art result at the time of publication.

Devlin et al. (2018) make the observation that because Radford et al.'s GPT uses a regular language modelling objective during pre-training, the representations it learns are not bidirectional, which limits their representational power. Indeed, because the "traditional" objective of a language model is to predict the next word in a sentence based on the ones it's already seen, truly bidirectional models cannot be used for this task, as they have already seen the next word in a sentence at any time step, which makes predictions trivial and prevents them from learning anything.

While bidirectionality can be mimicked with the combination of two separate left-to-right and right-to-left LMs (as is done in ELMo), Devlin et al. argue that this approach is sub-optimal compared to a truly bidirectional model. For this rea-

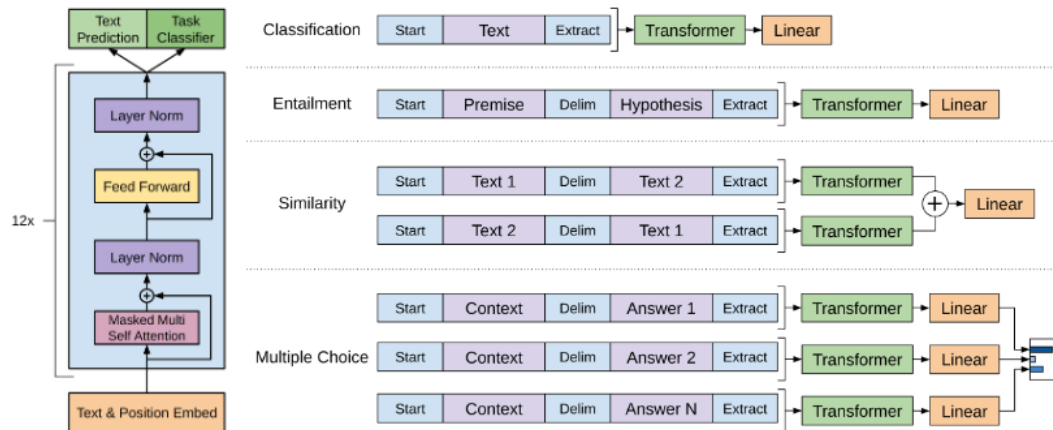


Figure 3.10: Architecture of the GPT and transfer to other tasks (Radford et al., 2018)

son, they introduce the *Bidirectional Encoder Representations from Transformers* (BERT), a fully bidirectional language model for transfer learning in NLU.

In order to make bidirectionality possible in BERT, Devlin et al. devise a new language modelling objective: the *Masked Language Model* (MLM). In the MLM, random words in a language model’s input are masked, and the goal is to predict them. This method makes bidirectionality possible, because models don’t know the words that are masked in advance, even if they have already seen them in the input. In addition to the MLM, Devlin et al. also introduce a next sentence prediction task during the pre-training of their model. In this task, pairs of sentence are extracted from the language model’s training data, with part of them following each other in the text, and others selected at random. The goal for the model is then to predict if the sentences it receives as input follow each other in the text or not.

For NLI, transfer learning with BERT is done as illustrated in figure 3.11 (taken directly from the original paper). The premise and hypothesis in a sentence pair are concatenated (with a separator token between them), and a special *class* ($[CLS]$) token is appended at the beginning of the sequence. For classification, an additional output layer is integrated at the end of the model to predict the relation between the sentences passed as input.

In order to make comparison with the work by Radford et al. possible, Devlin et al. propose a version of BERT with approximately the same number of parameters as the GPT, named $BERT_{BASE}$. In addition, another larger version of the model called $BERT_{LARGE}$ is trained and tested. Both significantly outperform all other existing models, and $BERT_{LARGE}$ provides impressive new state-of-the-art results which show the power of transfer learning from bidirectional language models to NLU.

Table 3.5 summarises the reported accuracies of the models presented in this section on SNLI and MultiNLI.

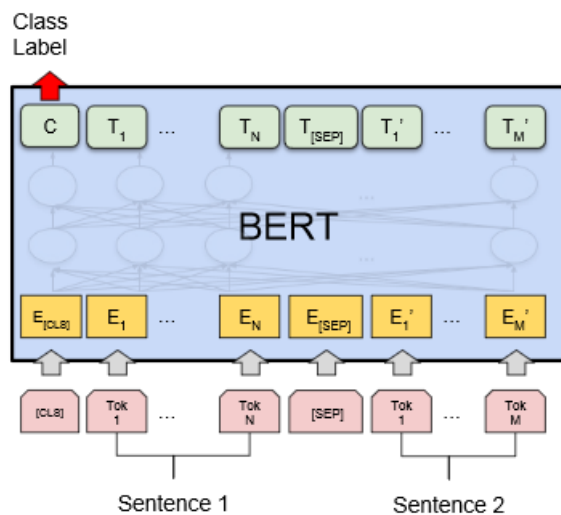


Figure 3.11: Transfer learning to NLI with BERT (Devlin et al., 2018)

Model	SNLI	MultiNLI-m	MultiNLI-mm
CoVe (McCann et al., 2017)	88.1	-	-
DMAN (B. Pan et al., 2018)	88.8	78.9	78.2
ELMo (Peters et al., 2018)	88.7	-	-
GPT (Radford et al., 2018)	89.9	82.1	81.4
<i>BERT</i> _{BASE} (Devlin et al., 2018)	-	84.6	83.4
<i>BERT</i> _{LARGE} (Devlin et al., 2018)	-	86.7	85.9

Table 3.5: Reported accuracy (%) of transfer learning approaches on SNLI and MultiNLI's matched (MultiNLI-m) and mismatched (MultiNLI-mm) test sets

Chapter 4

Conclusion

In this paper, we investigated existing approaches to the problem of natural language inference, an essential aspect of natural language understanding.

In the first chapter of this document, the crux of the problem was explained in detail and its importance was made clear through examples of applications where the ability to perform natural language inference is essential.

In chapter 2, we listed and detailed existing tasks and data sets for NLI. In particular, we saw that the release of the large scale, high quality SNLI corpus in 2015 by Bowman et al. paved the way for numerous deep learning approaches to natural language inference. We also noted how the wider coverage MultiNLI corpus pushed the quality of available data even further and was adopted as the new de facto standard to train and test models for recognising entailment. We finally mentioned how existing data sets were still imperfect and suffered from annotation artifacts that reduced the generalisation power of models trained on them.

In chapter 3, we thoroughly explored the different categories of models that exist in the literature for the task of recognising textual entailment. We saw how the wide range of linguistic phenomena covered by NLI made it an interesting task to learn sentence embeddings for other NLP tasks, which motivated the development of many sentence vector-based models trained on entailment data. We also underlined how the interactions between premises and hypotheses played an important role in recognising entailment, which justified the design of many sentence matching models based on attention mechanisms between pairs. Finally, we observed that recent approaches to NLI using transfer learning yielded impressive new state-of-the-art results on the task.

With all of the elements above in mind, we believe that further works on NLI should focus both on improving the definition of the task and on the design of new models to solve it.

In terms of task definition, new data sets with stricter rules for the writing of hypotheses could be developed in order to avoid annotation artifacts and to allow models to learn more general representations.

Regarding the design of new models, we have seen in this document's third chapter that only very little work had been done to investigate the use of external knowledge and lexical level information (i.e. lexical entailment) in deep learning models for NLI.

Further works should therefore try to include this type of information into existing models to assess if it can improve performance on the task.

Finally, with the excellent results obtained on existing data sets thanks to transfer learning from large scale language models, we expect that research on the topic of RTE will mostly focus on similar approaches in the near future.

Bibliography

- Bowman, Samuel R., Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts (2016). “A Fast Unified Model for Parsing and Sentence Understanding”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1466–1477. DOI: 10.18653/v1/P16-1139. URL: <https://www.aclweb.org/anthology/P16-1139>.
- Bowman, Samuel, Gabor Angeli, Christopher Potts, and Christopher D. Manning (2015). “A large annotated corpus for learning natural language inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Bromley, Jane, 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*, pp. 737–744.
- Chen, Qian, Zhen-Hua Ling, and Xiaodan Zhu (2018). “Enhancing Sentence Embedding with Generalized Pooling”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1815–1826. URL: <https://www.aclweb.org/anthology/C18-1154>.
- Chen, Qian, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei (2018a). “Neural Natural Language Inference Models Enhanced with External Knowledge”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2406–2417. URL: <https://www.aclweb.org/anthology/P18-1224>.
- (2018b). “Neural natural language inference models enhanced with external knowledge”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 2406–2417.
- Chen, Qian, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen (2017a). “Enhanced LSTM for Natural Language Inference”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1657–1668. DOI: 10.18653/v1/P17-1152. URL: <https://www.aclweb.org/anthology/P17-1152>.

- Chen, Qian, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen (2017b). “Recurrent Neural Network-Based Sentence Encoder with Gated Attention for Natural Language Inference”. In: *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 36–40. DOI: 10.18653/v1/W17-5307. URL: <https://www.aclweb.org/anthology/W17-5307>.
- Choi, Jihun, Kang Min Yoo, and Sang-goo Lee (2017). “Unsupervised Learning of Task-Specific Tree Structures with Tree-LSTMs”. In: *CoRR* abs/1707.02786. arXiv: 1707.02786. URL: <http://arxiv.org/abs/1707.02786>.
- Condori, Roque Enrique López and Thiago Alexandre Salgueiro Pardo (2017). “Opinion summarization methods: Comparing and extending extractive and abstractive approaches”. In: *Expert Systems with Applications* 78, pp. 124–134. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2017.02.006>. URL: <http://www.sciencedirect.com/science/article/pii/S0957417417300829>.
- Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes (2017). “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 670–680. DOI: 10.18653/v1/D17-1070. URL: <https://www.aclweb.org/anthology/D17-1070>.
- Conneau, Alexis, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov (2018). “XNLI: Evaluating Cross-lingual Sentence Representations”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.
- Dagan, Ido, Oren Glickman, and Bernardo Magnini (2006). “The PASCAL Recognising Textual Entailment Challenge”. In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*. Vol. 3944. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 177–190. DOI: 10.1007/11736790_9. URL: http://link.springer.com/10.1007/11736790_9.
- Das, Dipanjan and André FT Martins (2007). “A survey on automatic text summarization”. In: *Literature Survey for the Language and Statistics II course at CMU* 4, pp. 192–195.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805. arXiv: 1810.04805. URL: <http://arxiv.org/abs/1810.04805>.
- Ghaeini, Reza, Sadid A. Hasan, Vivek Datla, Joey Liu, Kathy Lee, Ashequl Qadir, Yuan Ling, Aaditya Prakash, Xiaoli Fern, and Oladimeji Farri (2018). “DR-BiLSTM: Dependent Reading Bidirectional LSTM for Natural Language Inference”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computa-

- tional Linguistics, pp. 1460–1469. DOI: 10.18653/v1/N18-1132. URL: <https://www.aclweb.org/anthology/N18-1132>.
- Giampiccolo, Danilo, Bernardo Magnini, Elena Cabrio, Hoa Trang Dang, Ido Dagan, and Bill Dolan (2008). “The Fourth PASCAL Recognizing Textual Entailment Challenge”. In: p. 11.
- Glockner, Max, Vered Shwartz, and Yoav Goldberg (2018). “Breaking NLI Systems with Sentences that Require Simple Lexical Inferences”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 650–655. URL: <https://www.aclweb.org/anthology/P18-2103>.
- Gong, Yichen, Heng Luo, and Jian Zhang (2018). “Natural Language Inference over Interaction Space”. In: *International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=r1dHXnH6->.
- Gururangan, Suchin, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith (2018). “Annotation Artifacts in Natural Language Inference Data”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 107–112. DOI: 10.18653/v1/N18-2017. URL: <https://www.aclweb.org/anthology/N18-2017>.
- Hirschman, Lynette, Marc Light, Eric Breck, and John D Burger (1999). “Deep read: A reading comprehension system”. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, pp. 325–332.
- Im, Jinbae and Sungzoon Cho (2017). “Distance-based Self-Attention Network for Natural Language Inference”. In: *CoRR* abs/1712.02047. arXiv: 1712.02047. URL: <http://arxiv.org/abs/1712.02047>.
- Kim, Seonhoon, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak (2018). “Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information”. In: *arXiv:1805.11360 [cs]*. arXiv: 1805.11360. URL: <http://arxiv.org/abs/1805.11360>.
- Lin, Zhouhan, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio (2017). “A Structured Self-attentive Sentence Embedding”. In: *CoRR* abs/1703.03130. arXiv: 1703.03130. URL: <http://arxiv.org/abs/1703.03130>.
- Liu, Pengfei, Xipeng Qiu, Yaqian Zhou, Jifan Chen, and Xuanjing Huang (2016). “Modelling Interaction of Sentence Pair with Coupled-LSTMs”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1703–1712. DOI: 10.18653/v1/D16-1176. URL: <https://www.aclweb.org/anthology/D16-1176>.
- Liu, Yang, Chengjie Sun, Lei Lin, and Xiaolong Wang (2016). “Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention”. In:

- CoRR* abs/1605.09090. arXiv: 1605.09090. URL: <http://arxiv.org/abs/1605.09090>.
- Marelli, M, S Menini, M Baroni, L Bentivogli, R Bernardi, and R Zamparelli (2014). “A SICK cure for the evaluation of compositional distributional semantic models”. In: p. 8.
- Marelli, Marco, 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)*. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). Dublin, Ireland: Association for Computational Linguistics, pp. 1–8. DOI: 10.3115/v1/S14-2001. URL: <http://aclweb.org/anthology/S14-2001>.
- McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher (2017). “Learned in Translation: Contextualized Word Vectors”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 6294–6305. URL: <http://papers.nips.cc/paper/7209-learned-in-translation-contextualized-word-vectors.pdf>.
- Miller, George A. (1995). “WordNet: A Lexical Database for English”. In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: <http://doi.acm.org/10.1145/219717.219748>.
- Mou, Lili, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin (2015). “Recognizing Entailment and Contradiction by Tree-based Convolution”. In: *CoRR* abs/1512.08422. arXiv: 1512.08422. URL: <http://arxiv.org/abs/1512.08422>.
- Munkhdalai, Tsendsuren and Hong Yu (2017). “Neural Semantic Encoders”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 397–407. URL: <https://www.aclweb.org/anthology/E17-1038>.
- Nie, Yixin and Mohit Bansal (2017). “Shortcut-Stacked Sentence Encoders for Multi-Domain Inference”. In: *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 41–45. DOI: 10.18653/v1/W17-5308. URL: <https://www.aclweb.org/anthology/W17-5308>.
- Pan, Boyuan, Yazheng Yang, Zhou Zhao, Yueting Zhuang, Deng Cai, and Xiaofei He (2018). “Discourse Marker Augmented Network with Reinforcement Learning for Natural Language Inference”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 989–999. URL: <http://aclweb.org/anthology/P18-1091>.
- Parikh, Ankur, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit (2016). “A Decomposable Attention Model for Natural Language Inference”. In: *Proceedings*

- of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics, pp. 2249–2255. DOI: 10.18653/v1/D16-1244. URL: <https://www.aclweb.org/anthology/D16-1244>.
- Pasunuru, Ramakanth, Han Guo, and Mohit Bansal (2017). “Towards improving abstractive summarization via entailment generation”. In: *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 27–32.
- Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). “Deep contextualized word representations”. In: *Proc. of NAACL*.
- Poliak, Adam, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme (2018). “Hypothesis Only Baselines in Natural Language Inference”. In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 180–191. DOI: 10.18653/v1/S18-2023. URL: <https://www.aclweb.org/anthology/S18-2023>.
- Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). “Improving language understanding by generative pre-training”. In:
- Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kociský, and Phil Blunsom (2015). “Reasoning about Entailment with Neural Attention”. In: *CoRR* abs/1509.06664. arXiv: 1509.06664. URL: <http://arxiv.org/abs/1509.06664>.
- Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei (2015). “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3, pp. 211–252. ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: <https://doi.org/10.1007/s11263-015-0816-y>.
- Sha, Lei, Baobao Chang, Zhifang Sui, and Sujian Li (2016). “Reading and thinking: Re-read LSTM unit for textual entailment recognition”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2870–2879.
- Shen, Tao, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang (2017). “DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding”. In: *CoRR* abs/1709.04696. arXiv: 1709.04696. URL: <http://arxiv.org/abs/1709.04696>.
- Shen, Tao, 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”. In: *CoRR* abs/1801.10296. arXiv: 1801.10296. URL: <http://arxiv.org/abs/1801.10296>.
- Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber (2015). “Training Very Deep Networks”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’15. Montreal, Canada:

- MIT Press, pp. 2377–2385. URL: <http://dl.acm.org/citation.cfm?id=2969442.2969505>.
- Talman, Aarne, Anssi Yli-Jyrä, and Jörg Tiedemann (2018). “Natural Language Inference with Hierarchical BiLSTM Max Pooling Architecture”. In: *CoRR* abs/1808.08762. arXiv: 1808.08762. URL: <http://arxiv.org/abs/1808.08762>.
- Tay, Yi, Anh Tuan Luu, and Siu Cheung Hui (2018). “Compare, Compress and Propagate: Enhancing Neural Architectures with Alignment Factorization for Natural Language Inference”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1565–1575. URL: <https://www.aclweb.org/anthology/D18-1185>.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 5998–6008. URL: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.
- Wang, Shuohang and Jing Jiang (2016). “Learning Natural Language Inference with LSTM”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 1442–1451. DOI: 10.18653/v1/N16-1170. URL: <https://www.aclweb.org/anthology/N16-1170>.
- Wang, Zhiguo, Wael Hamza, and Radu Florian (2017). “Bilateral Multi-perspective Matching for Natural Language Sentences”. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence. IJCAI’17*. Melbourne, Australia: AAAI Press, pp. 4144–4150. ISBN: 978-0-9992411-0-3. URL: <http://dl.acm.org/citation.cfm?id=3171837.3171865>.
- Williams, Adina, Nikita Nangia, and Samuel Bowman (2018). “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1112–1122. URL: <http://aclweb.org/anthology/N18-1101>.
- Yoon, Deunsol, Dongbok Lee, and SangKeun Lee (2018). “Dynamic Self-Attention : Computing Attention over Words Dynamically for Sentence Embedding”. In: *CoRR* abs/1808.07383. arXiv: 1808.07383. URL: <http://arxiv.org/abs/1808.07383>.
- Young, Peter, Alice Lai, Micah Hodosh, and Julia Hockenmaier (2014). “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions”. In: *Transactions of the Association for Computational Linguistics* 2, pp. 67–78. ISSN: 2307-387X. URL: <https://transacl.org/ojs/index.php/tacl/article/view/229>.

Zhang, Zhuosheng, Yuwei Wu, Zuchao Li, Shexia He, Hai Zhao, Xi Zhou, and Xiang Zhou (2018). “I Know What You Want: Semantic Learning for Text Comprehension”. In: *arXiv:1809.02794 [cs]*. arXiv: 1809.02794. URL: <http://arxiv.org/abs/1809.02794>.