DocNLI: A Large-scale Dataset for Document-level Natural Language Inference

Wenpeng Yin[1], Dragomir Radev[1,2] and Caiming Xiong[1]
[1]Salesforce Research and [2]Yale University
`wyin@salesforce.com`

Abstract

Natural language inference (NLI) is formulated as a unified framework for solving various NLP problems such as relation extraction, question answering, summarization, etc. It has been studied intensively in the past few years thanks to the availability of large-scale labeled datasets. However, most existing studies focus on merely sentence-level inference, which limits the scope of NLI's application in downstream NLP problems. This work presents DOCNLI — a newly-constructed large-scale dataset for document-level NLI. DOCNLI is transformed from a broad range of NLP problems and covers multiple genres of text. The premises always stay in the document granularity, whereas the hypotheses vary in length from single sentences to passages with hundreds of words. Additionally, DOCNLI has pretty limited artifacts[1] which unfortunately widely exist in some popular sentence-level NLI datasets. Our experiments demonstrate that, even without fine-tuning, a model pretrained on DOCNLI shows promising performance on popular sentence-level benchmarks, and generalizes well to out-of-domain NLP tasks that rely on inference at document granularity. Task-specific fine-tuning can bring further improvements. Data, code and pretrained models can be found at `https://github.com/salesforce/DocNLI`.

1 Introduction

A fundamental challenge of natural language processing (NLP) lies in the variability of semantic expression, where the same meaning can be conveyed by, or inferred from, different pieces of text (Dagan et al., 2009). This phenomenon gives rise to the many-to-many mapping between textual expressions and meanings. Many NLP problems, such as information extraction, question answering, document summarization and machine translation, desire a system for this variability phenomenon so as to figure out that a particular meaning can be inferred from distinct text strings (Dagan et al., 2009). Natural language inference (a.k.a textual entailment (Dagan et al., 2005)) acts as a unified framework to study those NLP problems by casting the background text as a premise and the text of target meaning as a hypothesis. Then, a good NLI recognizer can be considerably translated to a well-performing system regarding respective NLP tasks.

NLI was first studied in (Dagan et al., 2005). Research in the early stages was mostly driven by the PASCAL Recognizing Textual Entailment (RTE) challenges which are annual competitions with benchmark datasets released. In the past few years, the study of NLI has moved forward rapidly along with the construction of large-scale datasets, such as SNLI (Bowman et al., 2015), the science domain SciTail (Khot et al., 2018) and multi-genre MNLI (Williams et al., 2018), etc.

However, some NLI datasets may not be suitable any more for solving downstream NLP problems since they were commonly crowdsourced in isolation from any end task [2] (Khot et al., 2018). In addition, most NLI datasets and studies paid attention merely to sentence-level inference — both the premises and hypotheses are single (and usually short) sentences. This makes them unsuitable for other open-ended NLP problems. For example, to verify the factual correctness of a document summary, sentence-level NLI systems cannot be of much help (Kryściński et al., 2019). Considering the fact-checking task FEVER (Thorne et al., 2018) as another example, in order to figure out the truth value of a claim against a Wikipedia article, NLI

---

[1]NLI "artifacts" are some label-specific biases (often) in the hypotheses; they can indicate which NLI class a hypothesis belongs to even without looking at the premise.

[2]Except for RTE and SciTail

has to be done on individual sentences instead of using the whole article as the premise. In short, some NLP tasks require the reasoning of NLI to go beyond the sentence granularity, regarding both the premise and the hypothesis.

In this work, we introduce DOCNLI, a large-scale dataset for document-level NLI. It is constructed by reformatting some mainstream NLP tasks, including question answering and document summarization, and integrating existing NLI in which the premises may be longer than single sentences. DOCNLI has the following characteristics:

- DOCNLI is highly related with end NLP tasks. A well-performing system to DOCNLI is expected to throw light on addressing other NLP challenges.

- Premises always have more than one sentence; the majority are natural documents such as news articles. Hypotheses cover a variety of lengths, ranging from a single sentence to a document with hundreds of words. By this setting, we hope the systems can learn to deal with future applications that need to infer the truth value of a piece of text regardless of its length.

- In contrast to some existing sentence-level NLI datasets, DOCNLI has pretty limited artifacts. We present a novel approach to disconnect the potential artifacts with the NLI task itself; a "hypothesis-only" baseline has difficulties in discovering some spurious correlations.

In experiments, we will show that a RoBERTa (Liu et al., 2019) system pre-trained on DOC-NLI demonstrates promising performance on conventional sentence-level NLI benchmarks such as MNLI and SciTail, and generalizes well to out-of-domain NLP tasks (e.g., fact-checking and multi-choice question answering) that necessitate document-level inference. Task-specific fine-tuning can further improve the performance and achieve new state of the art for some end tasks.

2 Related Work

To our knowledge, document-level NLI has attracted very little ink in the community, possibly because of the lack of labeled datasets. In this section, we mainly describe some prior NLI datasets that share some spirits with our DOCNLI.

End-task driven. As mentioned in Section 1, the RTE series were driven by downstream NLP

tasks such as information retrieval, information extraction, question answering, and summarization. MCTest (Richardson et al., 2013) is a question answering task in which a paragraph is given as background knowledge, then each question is paired with a positive answer and some negative answers. The MCTest benchmark released an NLI version of this corpus by treating the whole paragraph as a premise and combining the question and answer candidates as hypotheses. SciTail (Khot et al., 2018) is also derived from the end QA task of answering multiple-choice school-level science questions. Unlike MCTest, the premises in SciTail are single sentences selected by an information retrieval approach. By casting an end NLP task as NLI, a good NLI recognizer therefore can be directly turned into a well-performing system for that NLP task. This can be even more attractive if we can learn a generalizable NLI system to solve some NLP problems that have limited annotations.

Going beyond the sentence granularity. The premises in MCTest are paragraphs, but MCTest has pretty limited size. Demszky et al. (2018) tried to convert the question answering benchmark SQuAD (Rajpurkar et al., 2016) into an NLI format by treating the paragraph as a premise and using a neural network to generate a hypothesis sentence given the question and answer span as inputs. Kryściński et al. (2019) created a (document, sentence) pair data "FactCC" to train a classifier for checking the factual correctness of single sentences in automatically generated summaries. FactCC is specific to the target summarization benchmark dataset, so it is unclear how well FactCC can generalize to other summarization benchmarks and other NLP problems. In addition, only single sentences act as hypotheses. Nevertheless, that literature exactly showed that document-level NLI, especially the inference of document-level hypotheses, is highly desirable. ANLI (Nie et al., 2020) also gather multi-sentence as premises. However, the sentence sizes in ANLI premises are pretty limited and the hypotheses in ANLI are single sentences consistently.

To our knowledge, our DOCNLI is the first dataset that uses hypotheses longer than single sentences, and stays closely with end NLP tasks.

3 Data Creation

What kind of document-level NLI dataset is preferred? (i) We want the premise is a paragraph or

	original task	domain	premise length	hypothesis length
ANLI	NLI	various (wiki, news, etc.)	multi-sentence (20~94 words)	single sentence (4~18 words)
SQuAD	QA	wiki	paragraph (27~237 words)	single sentence (6~22 words)
DUC (2001)	summarization	news	doc. (124~879 words)	multi-sent (80~100 words)
CNN/Daily Mail	summarization	news	doc. (247~652 words)	3~4 sent. (40~50 words)
Curation	summarization	news	doc. (229~842 words)	multi-sent (64~279 words)

Table 1: Data resources that are reformatted into DOCNLI.

even a document, and the hypotheses cover a large range of granularity: from a single sentence to a longer paragraph (e.g., 250 words); (ii) Diverse domains; (iii) No severe artifacts; for example, we do not include the hypotheses that can be easily found "grammatically incorrect" by well-trained language models such as BERT (Devlin et al., 2019).

3.1 Data Preprocessing

Table 1 lists all the resources that we use to create DOCNLI. Briefly, DOCNLI combines and reformats five existing NLP benchmarks: adversarial NLI (ANLI) (Nie et al., 2020), the question answering benchmark SQuAD (Rajpurkar et al., 2016) and three summarization benchmarks (DUC2001[3], CNN/DailyMail (Nallapati et al., 2016), and Curation[4] (Curation, 2020)). Next, we describe how each data resource is integrated into DOCNLI.

ANLI to DOCNLI. ANLI is a large-scale NLI dataset collected via an iterative, adversarial human-and-model-in-the-loop procedure. In each round, the best-performing model from the previous round is selected and then human annotators are asked to write "hard" examples that this model misclassifies. They always choose *multi-sentence paragraphs as premises and write single sentences as hypotheses*. Then a part of those "hard" examples join the training set so as to learn a stronger model for the next round. The remaining "hard" examples act as dev/test sets correspondingly. Totally three rounds were accomplished for ANLI construction. In the end, ANLI has train/dev/test

sizes as 162,865/3200/3200 with three classes "entail", "neutral" and "contradict".

We keep premise-hypothesis pairs in ANLI unchanged, but unify the two classes "neutral" and "contradict" into a new class "not_entail".

SQuAD to DOCNLI. SQuAD is a QA dataset in which a multi-sentence paragraph is accompanied by a couple of questions; each question has a text span from the paragraph as its answer. Demszky et al. (2018) converted SQuAD into NLI format by reformatting the question-answer pairs into declarative sentences (QA2D) by neural networks. The resulting sentences containing correct (resp. incorrect) answers are entailed (resp. not_entail) by the paragraph. Human evaluation was conducted to make sure those declarative sentences have high quality on three criteria: grammaticality, naturalness, and completeness. In addition, Demszky et al. (2018) replicated some statistical analyses showing that this QA2D dataset does not have clear artifacts as SNLI or MNLI. In this work, we directly use this QA2D dataset and re-split it into train/dev/test by 50k/7,236/8,275.

Summarization to DOCNLI. Here we introduce the basics of the three summarization datasets (DUC2001, CNN/DailyMail and Curation), and explain how we convert them into DOCNLI in a unified approach.

• The DUC series are some of the earliest benchmarks for studying automatic document summarization. DUC2001 is on generic, single-document summarization in the news domain. There are totally 600 documents along with human-written reference summaries of approximately 100 words. We split those document-summary pairs into train/dev/test by size of 400/50/150.

---

[3] https://www-nlpir.nist.gov/projects/duc/guidelines/2001.html
[4] https://github.com/CurationCorp/curation-corpus

doc	Petrofac shares surged on Wednesday following reports that the Serious Fraud Office has abandoned a criminal investigation into three businessmen who were accused of paying brides in the energy industry. The SFO had been probing claims that Unaoil - a Monaco-based consultancy that worked with Petrofac, primarily in Kazakhstan between 2002 and 2009 - had paid multimillion pound brides to land contracts in the oil and gas industry. But The Guardian cited sources earlier as saying that the SFO has dropped the investigation into the trio. Compliance industry newsletter MLex was the first to report the news, saying on Tuesday that the probe had been halted after three years. The SFO launched an investigation into Petrofac in May 2017 as part of a wider probe into Unaoil. In February 2019, David Lufkin, Petrofac's former global head of sales, pleaded guilty to 11 counts of bribery linked to contracts worth more than $730m in Iraq and $3.5bn in Saudi Arabia. SFO spokesman Adam Lilley said the Unaoil investigation "remains active and is ongoing". "We do not comment on ongoing investigations," he said. [···]
real summ.	The Serious Fraud Office has reportedly dropped a criminal investigation into three businessmen who had been accused of conspiring to make corrupt payments to secure contracts in Iraq. The SFO launched an investigation into Petrofac in May 2017 as part of a wider probe into Monaco-based oil consultancy Unaoil.
word repl.	The Serious financial Office has reportedly launched a criminal investigation into three businessmen who had been accused of conspiring to make corrupt payments to oil contracts in Iraq. The SFO launched an investigation into corruption in May 2017 as part of a wider investigation into Monaco-based financial consultancy firms.
entity repl.	Unaoil has reportedly dropped a criminal investigation into three businessmen who had been accused of conspiring to make corrupt payments to secure contracts in Monaco . The SFO launched an investigation into Monaco in May 2017 as part of a wider probe into Petrofac-based oil consultancy The Serious Fraud Office.
sent repl.	The Serious Fraud Office has reportedly dropped a criminal investigation into three businessmen who had been accused of conspiring to make corrupt payments to secure contracts in Iraq. A spokesman for the SFO said it was "unable to confirm or deny" that an inquiry had taken place.

Table 2: An example of the Curation summarization dataset shows the original document, and the real summary written by humans. We used "word replacement", "entity replacement" and "sentence replacement" to form three types of "fake" summaries against the document. Texts in red are substitutes.

- CNN/DailyMail was gathered from news articles in *CNN* and *Daily Mail* websites; each article is paired with 3 to 4 sentences of abstractive summary bullets generated by humans. CNN/DailyMail has 286,817/13,368/11,487 article-summary pairs in train/dev/test. The source articles in the training set have 766 words spanning 29.74 sentences on average while the summaries consist of averagely 3.72 sentences.

- Curation is a recent summarization dataset with 40,000 professionally-written summaries of news articles. We split it into train/dev/test as 20K/7K/13K.

All three summarization datasets align the documents with the human-written reference summaries. This enables us to obtain "entail" pairs of (document, reference summary). The remaining challenge lies in how to generate "not_entail" pairs.

We adopt three types of manipulations on the "reference" (also referred as "real") summaries.

- *Word replacement*. We mask eight words whose part-of-speech tags are among {"VERB", "NOUN", "PROPN", "NUM"} by spaCy toolkit[5], then use BERT to predict them. The most likely predicted word is used to replace a masked one. After word replacements, the resulting text is our "fake" summary.

- *Entity replacement*. We use spaCy for named

entity recognition (NER). For an entity which is the only one of a specific NER type in the real summary, we search for a different entity with the same type from the document to replace it; otherwise, it will be replaced by the entity of the same type in the real summary. We do this operation for five entities. We skip entity-level manipulation for the instances that have fewer than five detected entities. After entity replacement, we get a "fake" summary.

- *Sentence replacement*. From the real summary, we randomly select a sentence, then forward its left context to CTRL (Keskar et al., 2019), a state-of-the-art controllable text generator, to generate a new sentence which is used to replace the selected sentence. This operation generates a new "fake" summary.

Table 2 illustrates a (document, real summary) pair in the Curation dataset, and the three types of "fake" summaries we generated.

3.2 Mitigating Artifacts in DOCNLI

In Section 3.1, we transformed these NLI, QA and summarization datasets to satisfy the format of DOCNLI. We refer this resulting dataset as *raw-*DOCNLI. In consideration of the common artifacts in some popular sentence-level NLI benchmarks (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018), we tried a "hypothesis-only" baseline based on RoBERTa on this *raw-*DOCNLI. Sur-

---
[5] https://spacy.io

		entail	not_entail
DocNLI	raw DocNLI	ANLI SQuAD $\{(D, R)\}$	ANLI SQuAD $\{(D, F_i)\}$
	added pairs	$\{(F_i^+, F_i)\}$	$\{(F_i, R)\}$

Table 3: D: a document in summarization benchmarks; R: a real summary; F_i: a fake summary derived from R ($i=1\cdots n$); F_i^+: Using CTRL to insert a generated sentence between a random pair of consecutive sentences in the F_i, in a way similar to what we described as "sentence replacement" in Section 3.1. DocNLI's training set is the combination of raw-DocNLI and those added pairs; DocNLI's dev and test sets *do not have trivial pairs* $\{(F_i^+, F_i)\}$.

	train	dev	test
entail.	466,653	28,890	33,128
not_entail	475,661	205,368	233,958
sum	942,314	234,258	267,086

Table 4: Data sizes of DocNLI.

Figure 1: #premise vs. #words in DocNLI

prisingly, this baseline indeed obtains non-trivial performance. This means that RoBERTa can still learn some label-specific biases from the hypotheses, even though we have tried hard to make the "fake" summaries coherent and natural.

Nevertheless, this does not mean we have failed to build a robust DocNLI dataset. The surprising behavior of "hypothesis-only" in *raw*-DocNLI indicates that the BERT classifier can easily recognize the summary is "real vs. fake", but "real vs. fake" is not the same concept as "entail vs. not_entail" defined in the NLI framework. This is because a "fake" one can still be "entail"-ed if the premise has proper information; and a "real" one can also be "not_entail" if the premise does not contain necessary clues for inferring it.

For convenience, we use D as a document, R as the real summary, and $\{F_1, F_2, \cdots, F_n\}$ as the n fake summaries derived from R. To ensure the model can learn exactly what "entail vs. not_entail" is rather than be misled by the manipulations that yield those "fake" text pieces, as Table 3 demonstrates, we prepare the following pairs to extend the *raw*-DocNLI and get our final DocNLI:

- Adding pairs (F_i^+, F_i), $i = 1, \cdots, n$, for class "entail". Here F_i^+ has one more sentence than F_i, inserted by CTRL, as described in "sentence replacement" in Section 3.1 (here we do insertion rather than replacement). The goal is to let the system know that a fake summary can also be a positive hypothesis in NLI, if its premise covers necessary information.

- Adding a single pair (F_i, R) for class "not_entail". This means the original real summary can also be a negative hypothesis if
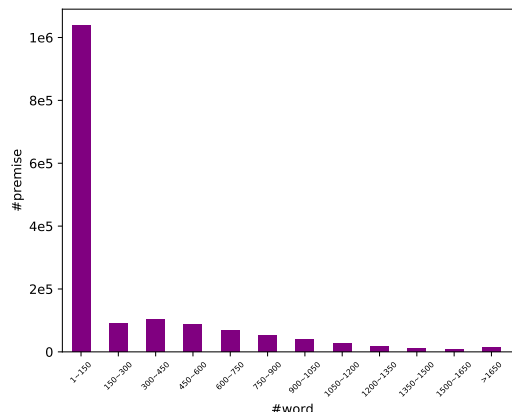
it includes mis-matching information with its premise. F_i is randomly chosen from the set $\{F_1, \cdots, F_n\}$.

By adding above two sorts of pairs, we want to disconnect the concept of "real vs. fake" from "entail vs. not_entail", letting the system learn the essence of NLI. Both the "real" and "fake" summaries have the same number of instances of being "entail" and "not_entail" in the extended dataset.

It is worth mentioning that since the instances "$(F_i^+, F_i) \rightarrow entail$" *are very trivial to be recognized, we add them in the training set only.*

Table 4 lists the sizes of DocNLI for train/dev/test in each class. The training set is roughly balanced, while approximately 12% examples in dev and test belong to "entail". F1 is the evaluation metric.

Figures 1-2 illustrate the length distributions of premises and hypotheses in DocNLI. Because the majority of hypotheses have fewer than 150 words, and real/fake summaries also act as premises in DocNLI, as reported in Table 3, therefore, the majority of premises stay within the length limit of 150 words, shown in Figure 1. Still, there are a large amount of premises whose lengths are within the range of [150, 900] words.

3.3 Human Verification

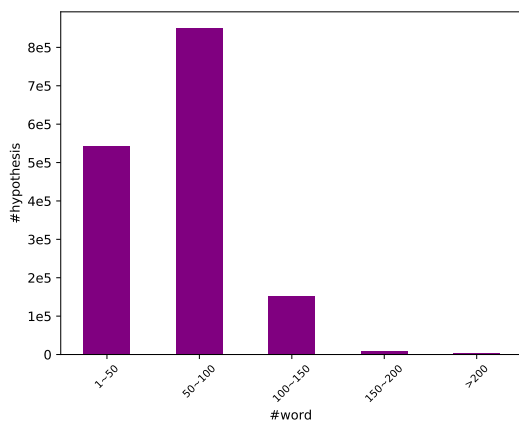DocNLI covers examples derived from ANLI, SQuAD and three summarization datasets. Here,

Figure 2: #hypothesis vs. #words in DOCNLI. The hypotheses in our new data DOCNLI are mostly longer than single sentences; this is one key difference with some related datasets.

Figure 3: Longformer F1 vs. #tokens in DOCNLI dev set.

	dev	test
Random	19.75	19.91
Hypothesis-only	21.89	22.02
Longformer-base	56.11	54.37
Roberta-large	64.06	61.52

Table 5: F1 scores on DOCNLI.

we only conduct human verification for the pairs derived from summarization, especially for those "fake" summaries, to get some clues to answer two questions: (i) Are those "fake" summaries indeed incorrect given the original document? (ii) Do those "fake" summaries look natural? By "natural" we mean the text should have no major grammar errors, and no unrelated text spans that make the whole text piece look over uncoordinated.

The authors of this work manually checked 200 random "fake" examples, among which none is true given the same document as the "real" summary. This is mainly because we replaced relatively a lot from the original real summaries.

However, some minor grammar issues inevitably exist. Take the following text piece as an example:

"*WeWork Companies LLC (replace: "We-Work") has announced plans to hold a conference call on 2025 for holders of its 7.875% Senior Notes due 26 August to discuss its Notes (replace: "Q2") results. Securities analysts and market-making financial institutions can also register for access. The call is scheduled for 12:00 P.M. (replace: "noon Eastern Time").*"

This example has five entities that are substitutes, all underlined. If a substitute comes from the premise document, we use "(replace: XX)" to denote the entity that was there. The two entities (NER type "date"), in red, replaced each other: "2025" and "26 August", which makes the new text "[···] on 2025 [···]" grammatically incorrect.
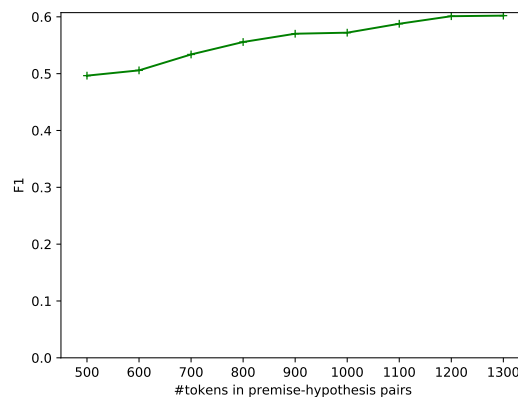
4 Experiments

We study three questions. (Q_1) How challenging is DOCNLI (especially with regard to different lengths of hypotheses)? (Q_2) Out-of-domain evaluation, in which we train a system given DOCNLI and test it on downstream NLP tasks that are not covered by the source tasks in DOCNLI construction. (Q_3) Could a system trained on DOCNLI work well on sentence-level NLI?

4.1 The DOCNLI task is challenging

The state-of-the-art systems on sentence-level NLI problems are largely based on transformers (Vaswani et al., 2017), such as BERT, RoBERTa (Liu et al., 2019), etc. However, they can only handle maximal 512 tokens preprocessed by the WordPiece tokenizer (Wu et al., 2016). This is an issue to build an effective document-level NLI machine. Therefore, for the main experiments, we also report Longformer (Beltagy et al., 2020) – a RoBERTa variant that can handle up to 4096 tokens. Longformer has two versions, one is "Longformer-base", the other is "Longformer-large". We currently only report "Longformer-base" due to memory constraints.

To answer the question (Q_1), we compare the following systems (we can include more baselines, but most popular approaches either are too weak or
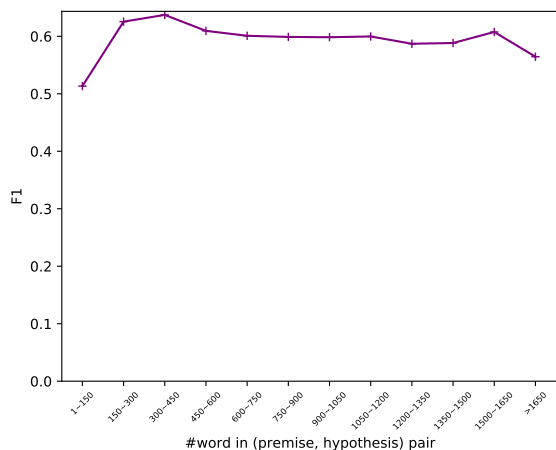
(a) F1 vs. pair length
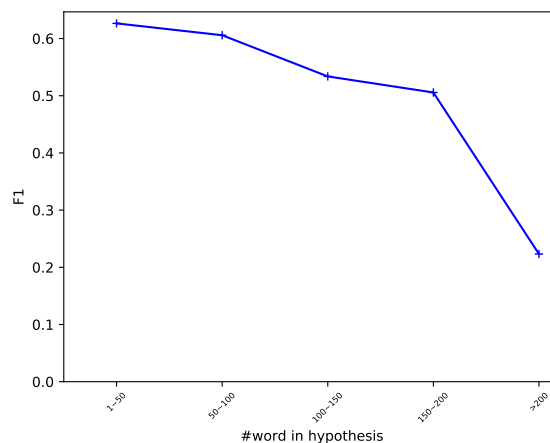
(b) F1 vs. hypothesis length

Figure 4: Fine-grained F1 scores for different lengths of DOCNLI pairs or hypotheses alone.

can only handle short piece of texts):

• **Hypothesis-only.** We train RoBERTa on hypotheses only.

• **RoBERTa-large.** Although we claimed that RoBERTa may not be a good platform to learn DOCNLI, here we report it just for reference. Maximal token limit: 512 tokens.

• **Longformer-base.** We use the released Longformer library[6] by (Beltagy et al., 2020), training it on the full training set of DOCNLI, with length limit of 1.3K tokens, batch size 1 per GPU, and learning rate 5e-6.

All systems are trained for 5 epochs, and report the best model tuned on dev set. Table 5 lists the F1 results of all systems on DOCNLI. We notice that "hypothesis-only" is just slightly higher than random guess, and is much lower than the "RoBERTa-large" system which takes both premises and hypotheses as input: 22.02 vs. 61.52 on test. Surprisingly, "Longformer"'s performance is clearly below that of the RoBERTa, even if it covers more tokens, possibly because we do not have enough computing resources to fully explore the better settings of Longformer. Figure 3 illustrates the impact of taking different numbers of tokens in Longformer, evaluated on dev set. In general, the more tokens the better performance.

We further look at the fine-grained F1 reports on the various lengths of premise-hypothesis pairs and hypotheses alone. Figure 4(a) shows that the system performance for pairs of lengths > 450 does not change clearly. This is probably due to those models' truncation when the (premise, hy-

pothesis) pairs are overlong (note that one word may be split into multiple tokens by the WordPiece tokenizer). Figure 4(b) demonstrates that the task gets increasingly challenging when the hypotheses become longer, which matches our intuition.

Overall, DOCNLI is a very challenging task that seeks solutions equipped with a stronger capability of representation learning.

4.2 Applying DOCNLI to end NLP tasks

To answer the question (**Q**₂), we play DOCNLI to see if it can help downstream NLP tasks. As DOCNLI is derived from summarization and QA already, we do not consider these two types of NLP tasks any more (since improvements on them are not surprising), especially when their domains are covered in DOCNLI. In addition, we have to explore tasks that have NLI-format data available — converting an open NLP task to NLI format is not trivial and is beyond the scope of this work. Therefore, we consider the following two NLP tasks:

FEVER (Thorne et al., 2018). FEVER is a benchmark dataset for fact-checking. Given an declarative sentence (aka. "claim"), the task searches for textual evidences from Wikipedia articles and then decide the truth value of this sentence (i.e., support / refute / not-enough-info).

We use the NLI-version of FEVER, released by (Nie et al., 2019): claims are hypotheses; premises corresponding to "support" or "refute' claims consist of ground truth textual evidence and some other randomly sampled evidence; premises for "not-enough-info" claims are the concatenation of all selected evidential sentences by a previ-

---

[6] https://github.com/allenai/longformer

	FEVER	MCTest	
	binary	v160	v500
random	50.00	25.00	25.00
pretrain MNLI	86.64	75.41	70.66
pretrain ANLI	87.51	82.50	78.66
pretrain DOCNLI	88.84	90.00	85.83
+finetune	**89.44**	**90.83**	**90.66**
Prior state-of-the-art	–	80.00	75.50

Table 6: Train on DOCNLI, test on NLP tasks that are out-of-domain and require document-level NLI. SOTA of MCTest comes from (Yu et al., 2019).

	SciTail	b-MNLI
majority	60.33	66.66
ESIM (Chen et al., 2017)	70.60	–
De-Att (Parikh et al., 2016)	72.30	–
DGEM (Khot et al., 2018)	77.30	–
BERT-large	89.71	90.55
Longformer-base	92.23	92.03
RoBERTa-large	95.13	93.95
DOCNLI (pretrain)	78.17	91.13
+finetune	96.04	94.07
Prior state-of-the-art	97.70	–

Table 7: Train on DOCNLI, test on sentence-level NLI benchmarks with or without fine-tuning. The SOTA of SciTail was reported by the DeBERTa model (He et al., 2020).

ous SOTA fact-checking system. We combine "refute" and "not-enough-info" as a single class "not_entail", and rename this data as "FEVER-binary". We randomly split FEVER-binary by 203,152/8,209/10,000 for train/dev/test respectively.[7]

MCTest (Richardson et al., 2013). In Related Work, we have introduced MCTest. Briefly, it is a multi-choice QA benchmark in the domain of **fictional story**. The authors of MCTest released an NLI-version MCTest by combining the question and the positive (resp. negative) answer candidate as a positive (resp. negative) hypothesis.

MCTest consists of two subsets. MCTest-160 contains 160 items (70 train, 30 dev, 60 test), each consisting of a document, four questions followed by one correct answer and three incorrect answers and MCTest-500 500 items (300 train, 50 dev, 150 test). *MCTest has pretty limited labeled data; thus, it is a good testbed to investigate* DOCNLI *in studying annotation-scarce tasks.* The MCTest has two official metrics: accuracy and NDCG (Normalized Discounted Cumulative Gain). Here we only report accuracy.

In this section, we still use RoBERTa-large and compare our DOCNLI with a latest NLI dataset ANLI in which the premises are longer than single sentences, and MNLI, the most widely-used sentence-level NLI dataset. For each data set (i.e., MNLI, ANLI or DOCNLI), we try two settings: (i) Using the data for pre-training, then do inference on FEVER-binary or MCTest directly without task-specific fine-tuning; (ii) First pre-training on the data, then fine-tune on FEVER-binary or MCTest.

In Table 6, DOCNLI can consistently generalize better than ANLI and MNLI on the two NLP tasks FEVER-binary and MCTest. We notice that the pretrained model on DOCNLI demonstrates very strong performance on the two end tasks, even without any fine-tuning on the task-specific examples. Especially for MCTest, both the "DOCNLI (pretrain)" and "DOCNLI+finetune" surpass the prior state-of-the-art by large margins.

4.3 Applying DOCNLI to sentence-level NLI

To answer the question (Q_3), we use SciTail and MNLI as target sentence-level NLI tasks. SciTail is from the science domain with two classes "entail" and "not_entail" split 23,596/1,304/2,126 (train/dev/test). MNLI covers a broad range of genres with three classes "entail/neutral/contradict" split 392,702/20k/20k (train/dev/test). Since the gold labels of the test set in MNLI are not publicly available and DOCNLI is a binary classification task, we first unify MNLI's "neutral" and "contradict" into "not_entail", then build a new labeled test set by randomly sampling 13k from the original dev set (the remaining examples are the new dev set). So now we have train/dev/test of size 372,702/6,647/13k. We first try some popular Transformer-style models, such as BERT-large, RoBERTa-large and Longformer-base to check how much we can get by training a supervised system on the full training data. Afterwards, we build a classifier by training RoBERTa-large on DOCNLI with or without SciTail/MNLI-specific fine-tuning.

Table 7 shows that: (i) The pretrained model on DOCNLI indeed can generalize to some extend on both SciTail and MNLI. In particular, it gets

---

[7]Please note that this data released by (Thorne et al., 2018) is different from the one used in FEVER leaderboard.

SciTail accuracy 78.17 which is even higher than some task-specific fully-supervised models such as "ESIM", "De-Att" and "DGEM". The same pretrained system can also get comparable performance with BERT, Longformer and RoBERTa on binary-MNLI; this should be attributed to the strong generalization of ANLI towards MNLI (Nie et al., 2020); (ii) When do task-specific fine-tuning, our model can further improve the performance and get very close to the state-of-the-art in SciTail.

5 Summary

In this work, we collect and release a large-scale document-level NLI dataset DOCNLI. It covers multiple genres and multiple ranges of lengths in both premises and hypotheses. We expect this dataset can help to solve some NLP problems that require document-level reasoning such as QA, summarization, fact-checking etc. In experiments, we show that DOCNLI can yield a model generalizing well to downstream NLP tasks and some popular sentence-level NLI tasks.

Acknowledgments

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *ACL*, pages 1657–1668.

Curation. 2020. Curation corpus base.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches - erratum. *Nat. Lang. Eng.*, 16(1):105.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop*, pages 177–190.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT*, pages 107–112.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTaiL: A textual entailment dataset from science question answering. In *AAAI*, pages 5189–5197.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *CoRR*, abs/1910.12840.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, pages 280–290.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *AAAI*, pages 6859–6866.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP*, pages 2249–2255.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In **SEM@NAACL-HLT* , pages 180–191. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, pages 193–203.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, pages 809–819.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *LREC*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, pages 1112–1122.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Jianxing Yu, Zhengjun Zha, and Jian Yin. 2019. Inferential machine comprehension: Answering questions by recursively deducing the evidence chain from text. In *Proceedings of ACL*, pages 2241–2251.