Combining Fact Extraction and Verification with Neural Semantic Matching Networks

Yixin Nie, Haonan Chen, Mohit Bansal
Department of Computer Science
University of North Carolina at Chapel Hill
{yixin1, haonanchen, mbansal}@cs.unc.edu

Abstract

The increasing concern with misinformation has stimulated research efforts on automatic fact checking. The recently-released FEVER dataset introduced a benchmark fact-verification task in which a system is asked to verify a claim using evidential sentences from Wikipedia documents. In this paper, we present a connected system consisting of three homogeneous neural semantic matching models that conduct document retrieval, sentence selection, and claim verification jointly for fact extraction and verification. For evidence retrieval (document retrieval and sentence selection), unlike traditional vector space IR models in which queries and sources are matched in some pre-designed term vector space, we develop neural models to perform deep semantic matching from raw textual input, assuming no intermediate term representation and no access to structured external knowledge bases. We also show that Pageview frequency can also help improve the performance of evidence retrieval results, that later can be matched by using our neural semantic matching network. For claim verification, unlike previous approaches that simply feed upstream retrieved evidence and the claim to a natural language inference (NLI) model, we further enhance the NLI model by providing it with internal semantic relatedness scores (hence integrating it with the evidence retrieval modules) and ontological WordNet features. Experiments on the FEVER dataset indicate that (1) our neural semantic matching method outperforms popular TF-IDF and encoder models, by significant margins on all evidence retrieval metrics, (2) the additional relatedness score and WordNet features improve the NLI model via better semantic awareness, and (3) by formalizing all three subtasks as a similar semantic matching problem and improving on all three stages, the complete model is able to achieve the state-of-the-art results on the FEVER test set (two times greater than baseline results).[1]

1 Introduction

The explosion of online textual content with unknown integrity and verification raises an important concern about misinformation such as fake news, socio-political deception, and online rumors. This problem of misinformation could potentially produce uncontrollable and harmful social impacts, thus stimulating recent research efforts on leveraging modern machine learning techniques for automatic fact checking. The recent release of the Fact Extraction and VERification (Thorne et al. 2018) (FEVER) dataset not only provides valuable fuel for applying data-driven neural approaches on evidence retrieval and claim verification, but also introduces a standardized, benchmark task of the automatic fact checking. In this FEVER shared task, a system is asked to verify an input claim with potential evidence in about 5 million Wikipedia documents, and label it as "SUPPORTS", "REFUTES", or "NOT ENOUGH INFO" if the evidence can support, refute, or not be found for the claim, respectively. Fig. 1 shows an example of the task. The task is difficult in two aspects. First, accurate selection of potential evidence from a huge knowledge base, w.r.t. an arbitrary claim requires a thoughtful system design and results in a trade-off between retrieval performance and computational resources. Moreover, even with ground truth evidence, the verification sub-task of predicting the relation between evidence and the claim is still a long-existing open problem.[2]

Claim: Giada at Home was only available on DVD.

[wiki/Giada_at_Home]
Giada at Home is a television show hosted by Giada De Laurentiis. It first aired on October 18, 2008 on the Food Network.

[wiki/Food_Network]
Food Network is an American basic cable and satellite television channel that is owned by Television Food Network, G.P., a joint venture and general partnership between Discovery, Inc. (which owns 70% of the network) and Tribune Media (which owns the remaining 30%).

Label: Refutes

Figure 1: Example of FEVER task. Given the claim, the system is required to find evidential sentences in the entire Wikipedia corpus and label it as "SUPPORTS", "REFUTES", or "NOT ENOUGH INFO" (Thorne et al. 2018).

In this work, we propose a joint system consisting of three connected homogeneous networks for the 3-stage FEVER task of document retrieval, sentence selection, and claim verification and frame them as a similar semantic matching

[1]Code: https://github.com/easonnie/combine-FEVER-NSMN

[2]The task is often termed as natural language inference (NLI).

problem. In the document retrieval phase, the corresponding sub-module selects documents from the entire Wikipedia corpus by keyword matching and uses a neural semantic matching network for further document ranking. In the sentence selection phase, we use the same neural architecture trained with an annealed sampling method to select evidential sentences by conducting semantic matching between each sentence from retrieved pages and the claim. Finally, we build a neural claim verifier by integrating upstream semantic relatedness features (from the sentence selector) and injecting ontological knowledge from WordNet into a similar neural semantic matching network for natural language inference (NLI), and train it to infer whether the retrieved evidence supports or refutes the claim, or state that the evidence is not enough to decide the correctness of the claim.

Overall, our unified neural-semantic-matching model for fact extraction and verification, which includes a three-fold contribution: (1) Unlike traditional IR methods e.g., TF-IDF, in which queries and sources are matched in some vector space according to pre-designed terms and pre-calculated weightings, we explore the possibility of using a neural semantic matching network for evidence retrieval and show that by assuming no intermediate term representation, neural networks can learn their own optimal representation for semantic matching at the granularity of sentences and significantly outperform term-weighting based methods. We also show that external Pageview frequency information can provide comparable and complementary discriminative information w.r.t. the neural semantic matching network for document ranking. (2) In contrast to previous work, in which upstream retrieved evidence are simply provided to downstream NLI models, we combined the evidence retrieval module with the claim verification module, by adding semantic relatedness scores to the NLI models, and further improve verification performance by using additional semantic ontological features from WordNet. (3) Rather than depending on structured machine-friendly knowledge bases such as Freebase (Bollacker et al. 2008) and DBpedia (Auer et al. 2007), we formalize the three subtasks as a similar textual semantic matching problem and propose one of the first neural systems that are able to conduct evidence retrieval and fact verification using raw textual claims and sentences directly from Wikipedia as input, and achieves the state-of-the-art results on the FEVER dataset, which could serve as a new neural baseline method for future advances on large-scale fact checking.

2 Related Works

Open Domain Question Answering: Recent deep learning-based open domain question-answering (QA) systems follow a two-stage process including document retrieval, selecting potentially relevant documents from a large corpus (such as Wikipedia), and reading comprehension, extracting answers (usually a span of text) from the selected documents. Chen et al. (2017a) was the first to successfully applied this framework to open domain QA, obtaining state-of-the-art results on several QA benchmarks (Rajpurkar et al. 2016; Baudiš and Šedivỳ 2015; Miller et al. 2016; Berant et al. 2013). Following their work, Dhingra, Mazaitis, and Cohen (2017) introduced new benchmarks, Wang et al. (2018) extended the framework by adding feedback signals from comprehension to the upstream document retriever, and Kratzwald and Feuerriegel (2018) proposed to adaptively adjust the number of retrieved documents. FEVER shares the similar retrieval problem as open domain QA, while the end task is claim verification.

Information Retrieval: Recent success in deep neural networks has brought increasing interest in their application to information retrieval (IR) tasks (Huang et al. 2013; Guo et al. 2016; Mitra, Diaz, and Craswell 2017; Dehghani et al. 2017). Although IR tasks also look at sentence-similarity, as discussed in Guo et al. (2016), they are more about relevance-matching, in which the match of specific terms plays an important role. As the end goal of FEVER is verification, its retrieval aspect is more about sentences having the same semantic meaning, and therefore, we approach the problem via natural language inference techniques, instead of relevance-focused IR methods.

Natural Language Inference: NLI is a task in which a system is asked to classify the relationship between a pair of premise and hypothesis as either entailment, contradiction or neutral. Large annotated datasets such as the Stanford Natural Language Inference (Bowman et al. 2015) (SNLI) and the Multi-Genre Natural Language Inference (Williams, Nangia, and Bowman 2018) (Multi-NLI) have promoted the development of many different neural NLI models (Nie and Bansal 2017; Conneau et al. 2017; Parikh et al. 2016; Chen et al. 2017b; Gong, Luo, and Zhang 2017; Ghaeini et al. 2018) that achieve promising performance. The task of NLI is framed as a typical semantic matching problem which exists in almost all kinds of NLP tasks. Therefore, in addition to the final claim verification subtask, we also formalize the other two FEVER subtasks of document retrieval and sentence selection as a similar problem and solve them using a homogeneous network.

Other FEVER Systems: FEVER is the first task that requires and measures models' joint ability for both evidence retrieval and verification, and provides a benchmark evaluation. There are other proposed methods during the FEVER Shared Task in which our system, Yoneda et al. (2018) and Hanselowski et al. (2018) are the top three on the leaderboard. The most apparent differences between our system and other methods are: (1) our method uses a homogeneous semantic matching network to tackle all the subtasks while others utilize different models for different subtasks and (2) our vNSMN verification model takes the concatenation of all the retrieved sentences (together with their relatedness score) as input, while other systems use existing NLI models to compare each of the evidential sentences with the claim and then apply another aggregation module to merge all the outputs for final prediction.

3 FEVER: Fact Extraction and VERification

3.1 Task Formalization

FEVER (Thorne et al. 2018) is a comprehensive task in which a system is asked to verify an arbitrary claim with

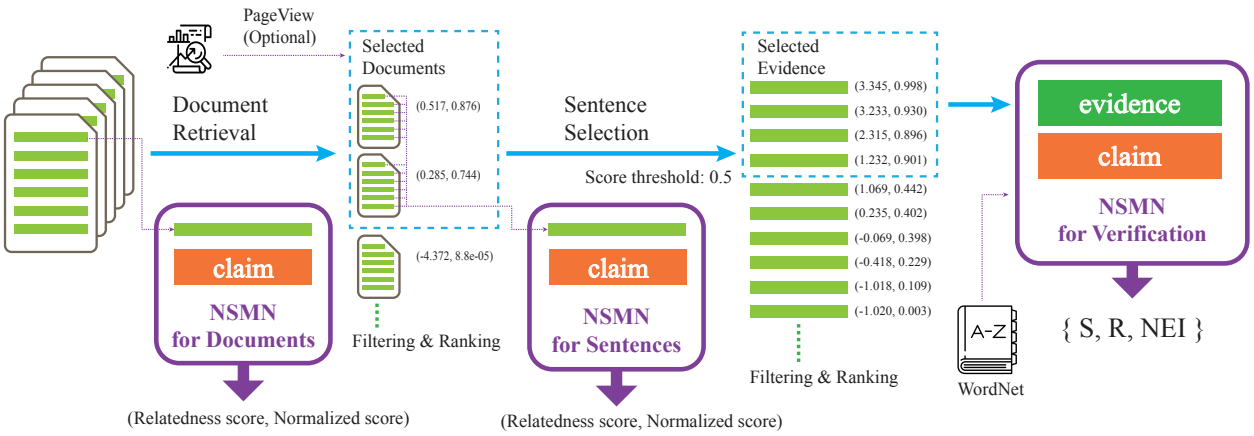

Figure 2: System Overview: Document Retrieval, Sentence Selection, and Claim Verification.

potential evidence extracted from a huge list of Wikipedia documents, or states that the claim is non-verifiable it cannot find enough evidence. Suppose $P_i \in \mathbb{P}$ denotes an individual Wikipedia document and $\mathbb{P} = \{P_0, P_1, ...\}$ is the set of all the provided documents. P_i is also an array of sentences, namely $P_i = \{s_i^0, s_i^1, s_i^2, ...s_i^m\}$ with each s_i^j denoting the j-th sentence in the i-th Wikipedia document (where s_i^0 is the title of the document). The inputs of each example are a textual claim c_i and $\bigcup P_i$, the union of all the sentences in each provided Wikipedia document. The output should be a tuple $(\hat{E}_i, \hat{y}_i)$ where $\hat{E}_i = \{s^{e_0}, s^{e_1}, ...\} \subset \bigcup P_i$, representing the set of evidential sentences for the given claim, and $\hat{y}_i \in \{S, R, NEI\}$[3], the predicted label for the claim. Suppose the ground truth evidence and label are E_i and y_i. For a successful verification of a given claim c_i, the system should produce a prediction tuple $(\hat{E}_i, \hat{y}_i)$ satisfying $E_i \subseteq \hat{E}_i$ and $y_i = \hat{y}_i$. The dataset details (size and splits) are discussed in Thorne et al. (2018). As described above, the single prediction is considered to be correct if and only if both the label is correct and the predicted evidence set (containing at most five sentences[4]) covers the annotated evidence set. This score is named as **FEVER Score**.

4 Our Model

In this section, we first describe the architecture of our Neural Semantic Matching Network (NSMN), and then elaborate on the three subtasks of document retrieval, sentence selection and claim verification, especially how these different sub-tasks can be treated as a similar semantic matching problem and be consistently resolved via the homogeneous NSMN architecture. See Fig. 2 for our system's overview.

4.1 Neural Semantic Matching Network

The Neural Semantic Matching Network (NSMN) is the key component in each of our sub-modules that performs semantic matching between two textual sequences. Specifically, the NSMN contains four layers as described below.[5]

Encoding Layer: Suppose the two input sequences are $\mathbf{U} \in \mathbb{R}^{d_0 \times n}$ and $\mathbf{V} \in \mathbb{R}^{d_0 \times m}$, the encoding layer is one bidirectional LSTM that encodes each input token with its contexts:

$$\bar{\mathbf{U}} = \text{BiLSTM}(\mathbf{U}) \in \mathbb{R}^{d_1 \times n}, \quad (1)$$

$$\bar{\mathbf{V}} = \text{BiLSTM}(\mathbf{V}) \in \mathbb{R}^{d_1 \times m}, \quad (2)$$

where d_0 and d_1 are input and output dimensions of the encoding layer and n and m are lengths of the two sequences.

Alignment Layer: This produces an alignment between the two input sequences based on the encoding of tokens computed above. The alignment matrix is computed as:

$$\mathbf{E} = \bar{\mathbf{U}}^\top \bar{\mathbf{V}} \in \mathbb{R}^{n \times m}. \quad (3)$$

Each element e_{ij} in the matrix indicates the alignment score between i-th token in $\mathbf{U}$ and j-th token in $\mathbf{V}$. Then, for each input token, the model computes the relevant semantic content from the other sequence using the weighted sum of encoded tokens according to the normalized alignment score:

$$\tilde{\mathbf{U}} = \bar{\mathbf{V}} \cdot \text{Softmax}_{\text{col}}(\mathbf{E}^\top) \in \mathbb{R}^{d_1 \times n}, \quad (4)$$

$$\tilde{\mathbf{V}} = \bar{\mathbf{U}} \cdot \text{Softmax}_{\text{col}}(\mathbf{E}) \in \mathbb{R}^{d_1 \times m}, \quad (5)$$

where $\text{Softmax}_{\text{col}}$ denotes column-wise softmax function. $\tilde{\mathbf{U}}$ is the aligned representation from $\bar{\mathbf{V}}$ to $\bar{\mathbf{U}}$ and vice versa for $\tilde{\mathbf{V}}$. The aligned and encoded representations are combined:

$$\mathbf{S} = f([\bar{\mathbf{U}}, \tilde{\mathbf{U}}, \bar{\mathbf{U}} - \tilde{\mathbf{U}}, \bar{\mathbf{U}} \circ \tilde{\mathbf{U}}]) \in \mathbb{R}^{d_2 \times n}, \quad (6)$$

$$\mathbf{T} = f([\bar{\mathbf{V}}, \tilde{\mathbf{V}}, \bar{\mathbf{V}} - \tilde{\mathbf{V}}, \bar{\mathbf{V}} \circ \tilde{\mathbf{V}}]) \in \mathbb{R}^{d_2 \times m}, \quad (7)$$

[3] S, R, NEI represent "SUPPORTS", "REFUTES" and "NOT ENOUGH INFO", respectively.

[4] This constraint is imposed in the FEVER Shared Task because in the blind test set, all claims can be sufficiently verified with at most 5 sentences of evidence.

[5] Our NSMN model is a modification of the Enhanced Sequential Inference Model (ESIM) (Chen et al. 2017b), where we add shortcut connections from input to matching layer and change output layer to only max-pool plus one affine layer with rectifier activation. These modifications are based on validation results.

where f is one affine layer with a rectifier as an activation function and $\circ$ indicates element-wise multiplication.

Matching Layer: The matching layer takes the upstream compound aligned representation and performs semantic matching between two sequences via a recurrent network as:

$$\mathbf{P} = \text{BiLSTM}([\mathbf{S}, \mathbf{U}^*]) \in \mathbb{R}^{d_3 \times n}, \quad (8)$$

$$\mathbf{Q} = \text{BiLSTM}([\mathbf{T}, \mathbf{V}^*]) \in \mathbb{R}^{d_3 \times m}. \quad (9)$$

Note that $\mathbf{U}^*$ and $\mathbf{V}^*$ are additional input vectors for each token provided to the matching layer via a shortcut connection. In this work, $\mathbf{U}^*$ and $\mathbf{V}^*$ are sub-channels of the input $\mathbf{U}$ and $\mathbf{V}$ without GloVe, aimed to facilitate the training.

Output Layer: The two matching sequences are projected onto two compressed vectors by max-pooling along the row axes. The vectors, together with their absolute difference and element-wise multiplication, are mapped to the final output $\mathbf{m}$ by a function h.

$$\mathbf{p} = \text{Maxpool}_{\text{row}}(\mathbf{P}) \in \mathbb{R}^{d_3}, \quad (10)$$

$$\mathbf{q} = \text{Maxpool}_{\text{row}}(\mathbf{Q}) \in \mathbb{R}^{d_3}, \quad (11)$$

$$h(\mathbf{p}, \mathbf{q}, |\mathbf{p} - \mathbf{q}|, \mathbf{p} \circ \mathbf{q}) = \mathbf{m}, \quad (12)$$

where function h denotes two affine layers with a rectifier being applied on the output of the first layer.

The final output vector is different for the extraction versus the verification subtasks. For the extraction subtasks (document retrieval and sentence selection), $\mathbf{m} = \langle m^+, m^- \rangle$, where $m^+ \in \mathbb{R}$ is a scalar value indicating the score for selecting the current sentence as evidence, and m^- gives the score for discarding it. For claim verification, $\mathbf{m} = \langle m_s, m_r, m_n \rangle$, where the elements of the vector denote the score for predicting the three labels, namely SUPPORTED, REFUTE, and NEI, respectively.

4.2 Three-Phase Procedure

1. Document Retrieval Document retrieval is the selection of Wikipedia documents related to a given claim. This sub-module handles the task as the following function:

$$f(c_i, \mathbb{P}) = D_{c_i}, \quad (13)$$

where a claim c_i and the complete collection of documents $\mathbb{P}$ are mapped to a set of indices D_{c_i} such that $\{P_i \mid i \in D_{c_i}\}$ is the set of documents required to verify the claim. The task can be viewed as selecting a fine-grained document subset from a universe of documents by comparing each of them with the input claim. We observe that 10% of the documents have disambiguation information in their titles, such as "Savages (band)"; the correct retrieval of these documents requires semantic understanding and we will rely on the NSMN to handle this problem. For clarity, we will call these documents as "disambiguative" documents for later use.

Because the number of documents in the collection are huge, conducting semantic matching between all the documents from the collection with the claim is computationally intractable. Hence, we start the retrieval by applying a **keyword matching** step to narrow down the search space.[6] The details about the keyword matching is described in the arxiv supplementary. Next, we give all the documents that are not "disambiguative" the highest priority score and rank the "disambiguative" documents by comparing each of them with the claim using the NSMN (Sec. 4.1). The document is represented as the concatenation of its title and the first sentence. The matching score between the claim c_i and the j-th document is computed as:

$$\langle m^+, m^- \rangle = \text{NSMN}(c_i, [t_j, s_j^0]), \quad (14)$$

$$p(x = 1 \mid c_i, j) = \frac{e^{m^+}}{e^{m^+} + e^{m^-}}, \quad (15)$$

where $x \in \{0, 1\}$[7] indicates whether to choose the j-th document, m^+ is the score for prioritizing the document, and $p(x = 1 \mid c_i, j) \in (0, 1)$ is the normalized score for m^+. Note that m^+ can also be viewed as the global semantic relatedness of a document to a claim. A document having a higher m^+ value than others, w.r.t. a claim, indicates that it is semantically more related to the claim.

To summarize, document retrieval phase involves steps:

- Building a candidate subset with keyword matching on all documents in the collections;
- Adding all the documents that are not "disambiguative" to the resulting list;[8]
- Calculating the $p(x = 1 \mid c_i, j)$ and m^+ value for "disambiguative" documents in the candidate set using NSMN;
- Filtering out the documents having $p(x = 1 \mid c_i, j)$ lower than some threshold P_{th}^d;
- Sorting the remaining documents by their m^+ values and adding the top k documents to the resulting list.

2. Sentence Selection Sentence selection is the extraction of evidential sentences from the retrieved documents regarding a claim, formalized as the following function:

$$g(c_i, \bigcup_{i \in D_{c_i}} P_i) = E_{c_i}, \quad (16)$$

which takes a claim and the union of the sentence set of each retrieved document as inputs and outputs a subset of sentences $E_{c_i} \subseteq \bigcup_{i \in D_{c_i}} P_i$ as the evidence set. Similar to document retrieval, sentence selection can also be treated as conducting semantic matching between each sentence $s_j \in \bigcup_{i \in D_{c_i}}$ and the claim c_i to select the most plausible evidence set. Since the search space is already narrowed down to a controllable size by the document retrieval, we can directly traverse all the sentences and compare them with the claim using NSMN. The selection is done via these steps:

- Calculating the $p(x = 1 \mid c_i, j)$ and m^+ value for all the sentences in the retrieved documents;
- Filtering out the sentences having $p(x = 1 \mid c_i, j)$ lower than some threshold P_{th}^s;
- Sorting sentences by their m^+ values and adding the top 5 sentences to the resulting list.

[6] On average, keyword matching returns 8 pages for each claim.

[7] 1 indicates to choose the document and 0 otherwise

[8] If there are multiple "disambiguative" documents, we randomly select at most five documents.

Exact same lemma
Antonym
Hyponym
Hypernym
Hyponym with 1-edge distance in WN topological graph
Hypernym with 1-edge distance in WN topological graph
Hyponym with 2-edge distance in WN topological graph
Hypernym with 2-edge distance in WN topological graph
Hyponym with distance > 2 edges in WN topological graph
Hypernym with distance > 2 edges in WN topological graph

Table 1: 10 indicator features in WordNet embedding.

3. Claim Verification This final sub-task requires logical inference from evidence to the claim, which is defined as:

$$h(E_{c_i}, c_i) = y, \tag{17}$$

where E_{c_i} is the set of evidential sentences and $y \in \{S, R, NEI\}$ is the output label.

We use the similar neural semantic matching network with additional token-level features for the final claim verification (similar to NLI task). The input premise is the concatenation of all sentences from the upstream evidence set, and the input hypothesis is the claim. More importantly, besides GloVe and ELMo[9] embeddings, the concatenation of the following three additional token-level features are added as task-specific token representations to further improve the verification accuracy:

WordNet: 30-dimension indicator features regarding ontological information from Wordnet. The 30 dimensions are divided into 10 embedding channels corresponding to 10 hypernymy/antonymy and edge-distance based phenomena, as shown in Table 1. For the current token, if one of these 10 phenomena is true for any word in the other sequence, that phenomenon's indicator feature will be fired. As we also want to differentiate whether the token is in the evidence or the claim, we use 3 elements for each channel with first two elements indicating the position and the last element for the feature indication. For example, if a token in the evidence fires then the vector will be [1, 0, 1] and if the token is in the claim it will be [0, 1, 1].

Number: We use 5-dimension real-value embeddings to encode any unique number token. This feature assists the model in identifying and differentiating numbers.

Normalized Semantic Relatedness Score: Two normalized relatedness scores, namely the two $p(x = 1 \mid c_i, j)$ values produced by the document and sentence NSMN, respectively. These scores are served as 2-dimension features for each token of the evidence. We add these features to connect the three subtask modules strongly, i.e., help the claim verification model better focus on the evidence, based on the semantic relatedness strength between the current evidence and the claim.

Evidence Enhancement (Optional): An optional step that augments the current evidence set. By default, we apply evidence enhancement before evaluation. Details are provided in the arxiv supplementary.

[9]We used GloVe+ELMo because their combination gives a comprehensive+contextualized lexical representation of the inputs.

5 Implementation and Training Details

Document Retrieval: The neural semantic matching network is trained by optimizing cross-entropy loss using the "disambiguative" documents containing ground truth evidence as positive examples and all other "disambiguative" document as negative examples. We used Adam optimizer (Kingma and Ba 2015) with a batch size of 128. We also consider using Pageview frequency resources, TF-IDF method after keyword matching for re-ranking the documents and compare their results in the experiments.

Sentence Selection: We trained neural sentence selector using the FEVER training set by optimizing cross-entropy loss with ground truth evidence as positive examples and all other sentences in the candidate pool from the document retriever as negative examples. We used Adam optimizer (Kingma and Ba 2015) with a batch size of 128. We applied an **annealed sampling** strategy to gradually increase the portion of positive examples after every epoch. Concretely, each negative example in the training data will be added to the next training epoch with a decreasing probability p_e. p_e starts from 0.5 at the first epoch and decreases 0.1 after each epoch and is reset to 0.02 when $p_e \leq 0$. The intuition behind annealed sampling is that we want the model to be more tolerant about selecting sentences while being discriminative enough to filter out apparent negative sentences. We also experiment with using TF-IDF method and a Max-Pool sentence encoder that gives a comprehensive representation for sentence modeling (Conneau et al. 2017) in place of the NSMN for sentence selection.

Claim Verification: We trained our verification NSMN using ground truth labels in FEVER training set. For verifiable claims, the input evidence is the provided ground truth supporting or refuting evidence. For non-verifiable claims with no given evidence, we randomly sample 3-5 sentences from candidate sentence pool with equal probability given by the upstream selected sentence. We use Adam optimizer for training the model with a batch size of 32.

6 Results and Analysis

In this section, we present extensive ablation studies for each module in our system, and report our final full-system results. When evaluating the performance of document retrieval and sentence selection, we compare the upper bound of the FEVER score (or oracle score **OFEVER**) by assuming perfect downstream systems.[10] Besides that, we also provide other metrics (i.e., F1 and label accuracy) for analyzing different submodules. For simplicity, we name dNSMN for document retrieval NSMN, sNSMN for sentence selection NSMN, and vNSMN for verification NSMN.

Document Retrieval Results In Table 2, we compare the performance of different methods for document retrieval on the entire dev set and on a difficult subset of the dev set. This subset is built by choosing examples having at least one evidence contained in the "disambiguative" document.

[10]**OFEVER** is the same metric as the "Oracle Accuracy" in the original baseline in (Thorne et al. 2018).

Model	Entire Dev Set				Difficult Subset (>10%)			
	OFEVER	Acc.	Recall	F1	**OFEVER**	Acc.	Recall	F1
FEVER Baseline	70.20	–	–	–	–	–	–	–
KM	88.86	44.90	83.30	58.35	60.15	23.89	60.15	34.20
KM + Pageview	91.98	45.90	87.98	60.32	85.61	29.32	85.61	43.68
KM + TF-IDF	91.63	42.83	87.45	57.50	85.60	28.66	85.60	42.94
KM + dNSMN	92.34	52.70	88.51	66.06	87.93	31.71	87.93	46.61
KM + Pageview + dNSMN	**92.42**	**52.73**	**88.63**	**66.12**	**88.73**	**31.90**	**88.72**	**46.93**
k = 5								
FEVER Baseline	77.24	–	–	–	–	–	–	–
KM	90.69	42.61	86.04	56.99	74.34	23.19	74.34	35.36
KM + Pageview	92.69	42.92	89.04	57.92	90.52	24.89	90.52	39.05
KM + TF-IDF	92.38	39.57	88.57	54.70	89.88	23.94	89.88	37.80
KM + dNSMN	**92.82**	51.04	**89.23**	**64.94**	91.33	28.30	91.33	43.21
KM + Pageview + dNSMN	92.75	**51.06**	89.13	64.93	**91.36**	**28.38**	**91.37**	**43.30**
k = 10								

Table 2: Performance of different document retrieval methods. ***k*** indicates the number of retrieved documents. The last four columns show results on the difficult subset that includes more than 10% of dev set. dNSMN = document retrieval Neural Semantic Matching Network. 'KM'=Keyword Matching.

Method	Entire Dev Set				Difficult Subset (>12%)			
	OFEVER	Acc.	Recall	F1	**OFEVER**	Acc.	Recall	F1
FEVER Baseline	62.81	–	–	–	–	–	–	–
TF-IDF	83.77	34.16	75.65	47.07	53.01	38.54	51.01	44.63
Max-Pool Enc.	84.08	59.52	76.13	66.81	73.68	54.13	73.68	62.41
sNSMN w/o AS	86.65	**69.43**	79.98	**74.33**	68.34	**67.82**	68.34	**68.08**
sNSMN w. AS	**91.19**	36.49	**86.79**	51.38	**81.44**	34.56	**81.44**	48.53

Table 3: Different methods for sentence selection on dev set. 'Enc.'= Sentence Encoder. 'AS'= Annealed Sampling. The **OFEVER** column shows Oracle FEVER Score. The other three columns show the evidence accuracy, recall, and F1.

We hypothesize that the correct retrieval of these documents will be more semantically demanding and challenging. To begin with, the keyword matching method (getting 88.86% and 90.69% oracle score for $k = 5$ and 10) is better than the FEVER baseline (getting 70.20% and 77.24% oracle score for $k = 5$ and 10) with TF-IDF in Chen et al. (2017a). This is due to the fact that keyword matching with only titles and claims (as described in Sec. 4.2) is intuitively more related to human online search behavior and can narrow the search space down with very high accuracy, whereas filtering document using term-based method e.g., TF-IDF directly on the entire document collection tends to impose more errors. However, keyword matching does not maintain its performance on the difficult subset and suffers a 25 points drop on the oracle score because the method is essentially semantics-agnostic, i.e. can not reason by taking linguistic context into consideration. This imposed difficulty is better handled by reranking based on Pageview frequency, TF-IDF, and the dNSMN, with the last one outperforming the other two on all the metrics. Though dNSMN and Pageview frequency ranking obtain comparable results on oracle score, the two methods are inherently different in that the former approaches document selection via advanced self-learned deep representations while the latter via demographic bias. Thus, we also experiments on combining the two methods by first reranking using Pageview and then dNSMN. Finally, though the performances of all the methods are affected by increasing the number of retrieved documents from 5 to 10, the methods that use dNSMN merely suffered a 1 point drop on the retrieval accuracy on the entire dev set, indicating that it's relatively more robust than other methods.

Sentence Selection Results Similar to the document retrieval setup, we evaluate the sentence selection performance on both the entire dev set and a difficult subset. The difficult subset for sentence selection is built by selecting examples in which the number of word-overlap between the claim and the ground truth evidence is below 2 and thus requires higher semantic understanding. Neural networks with better lexical representations are intuitively more robust at selecting semantically related sentences than term weighting based methods. This fact is reflected in Table 3, where although TF-IDF and the Max-pool Sentence Encoder obtain similar oracle FEVER scores (83.77% and 84.08%) and evidence recall (75.65% and 76.13%), the latter could achieve a much higher score for all metrics on the difficult subset. Note that for the entire dev set, the oracle score of the normally-trained (without annealed sampling) sNSMN (86.65%) is higher than that of the Max-Pool sentence encoder (84.08%) but on the difficult set, the sNSMN obtains

Model	FEVER	LA	F1
			S/R/NEI
Final Model	**66.14**	**69.60**	**75.7**/69.4/**63.3**
w/o WN and Num	65.37	68.97	74.7/68.0/63.3
w/o SRS (sent)	64.90	69.07	74.5/**70.7**/60.7
w. SRS (doc)	66.05	69.69	75.6/70.0/62.8
Vanilla ESIM	65.07	68.63	73.9/68.1/63.0
Data from sNSMN			
Final Model	62.48	67.23	72.6/70.4/56.3
Data from TF-IDF			

Table 4: Ablation study for verification (vNSMN). 'WN'=WordNet feature, 'Num'=number embedding, 'Final Model'=vNSMN with semantic relatedness score feature only from sentence selection. 'SRS (sent)', 'SRS (doc)' = Semantic Relatedness Score from document retrieval and sentence selection modules. **FEVER** column shows strict FEVER score and LA column shows label accuracy without considering evidence. The last column shows F1 score of three labels. All models above line are trained with sentences selected from sNSMN for non-verifiable examples, while model below is from TF-IDF.

Threshold	FEVER	LA	Acc.	Recall	F1
0.5	66.15	69.64	36.50	86.69	51.37
0.3	66.42	69.76	33.17	86.90	48.01
0.1	66.43	69.67	29.83	86.97	44.42
0.05	**66.49**	**69.72**	28.64	87.00	43.10

Table 5: Dev set results (before evidence enhancement) for a vNSMN verifier making inference on data with different degrees of noise, by filtering with different score thresholds.

a lower recall (68.34%) compared to Max-Pool sentence encoder (73.68%). This is due to the fact that the model with a stronger alignment mechanism will be more strict about selecting evidence and thus tends to trade accuracy for recall. This motivates our usage of annealed sampling in order to improve evidence recall. Although the annealed sampling reduces the evidence F1, we will explain later that this improvement of recall is important for the final FEVER Score.

Claim Verification Results We also conduct ablation experiments for the vNSMN with the best retrieved evidence[11] on the FEVER dev set. Specifically, we choose the vNSMN with semantic relatedness score feature only from sentence selection as our **Final Model** (because it obtains the best results on FEVER score), and make modifications based on that model for analyzing different add-ons. The results are included in Table 4. First of all, we see that WordNet features (WN) and number embedding (Num) is able to increase the FEVER score, specifically by improving roughly 1 point of F1 scores on both the "SUPPORTS" (from 74.7 to 75.7) and the "REFUTES" (from 68.0 to 69.4) examples because ontological features from WordNet (e.g., symptoms, antonyms, and hypernymms) and ordinal numeral features provide discriminative and fine-grained relational information which is extremely useful for revealing entailment and contradiction relations. More importantly, by incorporating the semantic relatedness score from the sNSMN model into the downstream vNSMN model, we also observe a 1 point improvement on FEVER score and almost 3 points improvement on F1 score for "NOT ENOUGH INFO" examples. This approach can be viewed as combining evidence extraction with verification, by providing the verifier the degree of trustworthiness for each evidence and helping it recognize subtle neural relations between evidence and the claim. We also see that the vNSMN with semantic relatedness score from both document retrieval and sentence selection modules achieves comparable (slightly worse) results to the vNSMN with semantic relatedness score from only the sentence selection module (hence, we use the latter for our final model). Our intuition of this phenomenon is that the document extraction subtask is two hops away from the claim verification subtask and hence its annotation supervision is less useful than the sentence selection subtask which is only one hop away. We also compare our vNSMN (66.14% on FEVER) with vanilla ESIM model (65.07% on FEVER) and the results on all metrics demonstrate that our architecture is better at modeling semantic matching. Lastly, we compare the performances of the same vNSMN with different training data for non-verifiable examples. The change of training data induces significant drops on both FEVER accuracy and F1 for the "Not Enough Info" example, highlighting the importance of the quality of upstream training data for neural inference model.

[11]The best retrieved evidence is extracted from our dNSMN and sNSMN models (trained with annealed sampling).

Combination	FEVER
Pageview + dNSMN + sNSMN + vNSMN	**66.59**
dNSMN + sNSMN + vNSMN	66.50
Pageview + sNSMN + vNSMN	66.43

Table 6: Performance of different combinations on dev set.

Model	F1	LA	FEVER
UNC-NLP (our shared task model)	52.96	68.21	64.21
UCL Machine Reading Group	34.97	67.62	62.52
Athene UKP TU Darmstadt	36.97	65.46	61.58
UNC-NLP (our final model)	52.81	68.16	**64.23**

Table 7: Performance of systems on blind test results.

Noise Tolerance of vNSMN We evaluate the robustness of the vNSMN to noisy evidence during inference by setting different probability thresholds for filtering upstream evidence where the default 2-way softmax classification threshold is 0.5. By reducing this value, we are allowing less confident evidence to be selected for downstream vNSMN. In Table 5, we can see that the overall FEVER score is slightly increasing with the decrease of the threshold, indicating that

the vNSMN is immune to noise. The findings encourage our usage of annealed sampling during sentence selection training and providing high recall evidence for the final fact verification model. We set threshold to 0.05 for sentence.

Combination Evaluation and Final Result Since the KM + dNSMN and KM + Pageview + dNSMN setups get similar results on document retrieval (see Table 2), we also compare their final FEVER results using the best downstream model (see Table 6). Based on these dev FEVER results, we choose our final model as the combination of Pageview and dNSMN for blind test evaluation (though the non-Pageview neural-only model is still comparable). Finally, in Table 7, we present blind test results of our final system together with the top 3 results on the FEVER Shared Task leaderboard[12]. Our final system (Pageview + dNSMN + sNSMN + vNSMN) is able to get comparable results with our earlier shared task rank-1 system (Pageview + sNSMN + vNSMN), achieving the new state-of-the-art on FEVER.

7 Conclusion

We addressed the fact verification FEVER task via a three-stage setup of document retrieval, sentence selection, and claim verification. We develop consistent and joint neural semantic matching networks for all three subtasks, along with Pageview, WordNet, and inter-module features, achieving the state-of-the-art on the task.

Acknowledgments

We thank the reviewers for their helpful comments, and support via Verisk, Google, and Facebook research awards.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer. 722–735.

Baudiš, P., and Šedivỳ, J. 2015. Modeling of the question answering task in the yodaqa system. In *CLEF*.

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. *EMNLP*.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017a. Reading wikipedia to answer open-domain questions. *ACL*.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017b. Enhanced lstm for natural language inference. In *ACL*.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. *EMNLP*.

Dehghani, M.; Zamani, H.; Severyn, A.; Kamps, J.; and Croft, W. B. 2017. Neural ranking models with weak supervision. In *SIGIR*.

Dhingra, B.; Mazaitis, K.; and Cohen, W. W. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Ghaeini, R.; Hasan, S. A.; Datla, V.; Liu, J.; Lee, K.; Qadir, A.; Ling, Y.; Prakash, A.; Fern, X.; and Farri, O. 2018. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. In *NAACL-HLT*.

Gong, Y.; Luo, H.; and Zhang, J. 2017. Natural language inference over interaction space. *ICLR*.

Guo, J.; Fan, Y.; Ai, Q.; and Croft, W. B. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*.

Hanselowski, A.; Zhang, H.; Li, Z.; Sorokin, D.; Schiller, B.; Schulz, C.; and Gurevych, I. 2018. Ukp-athene: Multi-sentence textual entailment for claim verification. *FEVER*.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. *ICLR*.

Kratzwald, B., and Feuerriegel, S. 2018. Adaptive document retrieval for deep question answering. In *EMNLP*.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. In *EMNLP*.

Mitra, B.; Diaz, F.; and Craswell, N. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*.

Nie, Y., and Bansal, M. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *RepEval*.

Parikh, A. P.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*.

Wang, S.; Yu, M.; Guo, X.; Wang, Z.; Klinger, T.; Zhang, W.; Chang, S.; Tesauro, G.; Zhou, B.; and Jiang, J. 2018. R3: Reinforced reader-ranker for open-domain question answering. *AAAI*.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

Yoneda, T.; Mitchell, J.; Welbl, J.; Stenetorp, P.; and Riedel, S. 2018. Ucl machine reading group: Four factor framework for fact finding (hexaf). *FEVER*.

[12] http://fever.ai/task.html