Language Resources and Evaluation (2023) 57:611–639
https://doi.org/10.1007/s10579-023-09635-6

ORIGINAL PAPER

A semantics-aware approach for multilingual natural language inference

Phuong Le-Hong[1] · Erik Cambria[2]

Accepted: 11 January 2023 / Published online: 14 February 2023

Abstract
This paper introduces a semantics-aware approach to natural language inference which allows neural network models to perform better on natural language inference benchmarks. We propose to incorporate explicit lexical and concept-level semantics from knowledge bases to improve inference accuracy. We conduct an extensive evaluation of four models using different sentence encoders, including continuous bag-of-words, convolutional neural network, recurrent neural network, and the transformer model. Experimental results demonstrate that semantics-aware neural models give better accuracy than those without semantics information. On average of the three strong models, our semantic-aware approach improves natural language inference in different languages.

Keywords Language inference · Semantics · Recurrent neural networks · Transformers · Commonsense · Text

1 Introduction

Many important problems in natural language processing (NLP) such as dialogue systems, information retrieval, semantic parsing, commonsense reasoning, depend on natural language understanding (NLU). The task of natural language inference (NLI) is well positioned to serve as a benchmark task for research on NLU (Williams et al., 2018). This task is also known as recognizing textual entailment (Bos & Markert, 2005; MacCartney & Manning, 2009).

✉ Phuong Le-Hong
phuonglh@vnu.edu.vn

Erik Cambria
cambria@ntu.edu.sg

[1] Vietnam National University, Hanoi, Vietnam

[2] School of Computer Science and Engineering, NTU, Singapore, Singapore

In the NLI task, a model is presented with a pair of sentences and is asked to judge the relationship between their meanings, typically *entailment*, *contradiction* or *independence* (or *neutral*). In order for a NLI model to be efficient, it must handle crucial linguistic phenomena like coreference, quantification, tense, modality, lexical entailment, and lexical and syntactic ambiguity. On the one hand, a good NLI model must be able to extract good representations for the meanings of sentences, notably their lexical and compositional semantics. On the other hand, it must succeed at one or more difficult machine learning problems like structure prediction and knowledge access.

Before 2015, the primary sources of annotated NLI corpus were the recognizing textual entailment (RTE) challenge tasks. There are generally high-quality, manually-labeled datasets but all have a small size, fewer than a thousand of examples each. The SemEval 2014 task, called Sentence Involving Compositional Knowledge (SICK), prepared a larger corpus of 4,500 training examples, which used a semi-automatic construction approach. The Stanford NLI corpus (SNLI, Bowman et al. (2015)) was introduced in 2015 is the first large-scale manually annotated corpus, containing 570K sentence pairs. This corpus has enabled a good deal of progress on NLU, especially on core representation learning techniques for sentence understanding. Table 1 shows some examples of sentence pairs and their annotations from the development section of the SNLI corpus.

The SNLI corpus has two main limitations. First, the sentences in SNLI are derived from only a single text genre—image captions. This makes that the curated sentences are typically short and cannot cover a wide range of important phenomena such as temporal reasoning, belief and modality. Second, SNLI was proved to be not sufficiently demanding to serve as an effective benchmark for NLU—the best machine learning model performance nearly reaches human accuracy, making fine-grained comparisons between strong models difficult. In 2018, the Multi-Genre NLI corpus (MultiNLI) was introduced, which improved upon the SNLI corpus in both its coverage and difficulty (Williams et al., 2018). This corpus has 433K sentence pairs, representing both written and spoken speech in a wide range of styles, degrees of formality and topics.

The MultiNLI corpus has allowed explicit evaluation of models both on the quality of their sentence representations within the training domain and on their ability to derive good representations in unfamiliar domains. However, all the ten different genres of this corpus are of written and spoken American English and models trained on this corpus cannot be directly used beyond English. In order to perform and evaluate cross-lingual language understanding (XLU), the development and test sets of the MultiNLI corpus were extended to 15 languages, including low-resource languages. This dataset is called XNLI. XNLI consists of 7500 human-annotated development and test examples in NLI three-way classification format, making a total of 112,500 annotated pairs (Conneau et al., 2018). In particular, the Vietnamese section of XNLI has never been used for supervised learning evaluation in a monolingual setting. It has been used to evaluate cross-lingual pretrained models for NLI.

In this paper, we make the following main contributions:

Table 1 Some examples of sentence pairs and their annotations from the SNLI corpus

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country	*contradiction* C C C C C	The man is sleeping
An older and younger man smiling	*neutral* N N E N N	Two men are smiling and laughing at the cats playing on the door
A black race car starts up in front of a crowd of people	*contradiction* C C C C C	A man is driving down a lonely road
A soccer game with multiple males playing	*entailment* E E E E E	Some men are playing a sport

- We present and compare four neural models for natural language inference and establish the first baseline on the Vietnamese dataset. The proposed models make use of different sentence encoders, including continuous bag-of-words, convolutional neural network, recurrent neural network, and the transformers network.
- We propose to integrate two important pre-processing steps, namely word segmentation and part-of-speech tagging, to split input sentences into lexical units so as to improve the performance of the proposed models.
- We propose a method to exploit and integrate lexical semantics information of ConceptNet into the neural models for NLI. Extensive experiments demonstrate that semantics-aware neural models give better accuracy than those without semantics information.
- In addition to Vietnamese, we extend our method to English where we perform evaluation on an English dataset and also obtain improvement. Our findings are potentially valid for many languages.
- Our code, embeddings and dictionaries are publicly available.[1]

The remainder of this paper is structured as follows. Section 2 presents related work, focusing on natural language inference and integration of knowledge into deep learning models. Section 3 describes the four different neural network models, namely bag-of-words models, sequential models, parallel models, and BERT models. Section 4 proposes a method for exploiting and integrating semantics information of ConceptNet into the neural network models to improve their performance. Section 5 presents experimental results on NLI datasets of Vietnamese and English. Finally, we provide concluding remarks and discuss future work in Sect. 6.

[1] https://github.com/phuonglh/vlp, under the *nli* module.

2 Related work

2.1 Natural language inference

Over the past two decades, NLI has been addressed using a variety of techniques, including those based on symbolic logic (Fyodorov et al., 2000; MacCartney & Manning, 2009; Bos & Markert, 2005), statistical methods (Giampiccolo et al., 2007; de Marneffe et al., 2008) and neural networks (Bowman et al., 2015; Liu et al., 2019). More recently, deep contextual language models have been shown effective for learning universal language representations, leveraging large amount of unlabeled data and obtaining very good results in many natural language processing tasks, including NLI. The best performing system has achieved the GLUE score (General Language Understanding Evaluation) of 90.7% (Wang et al., 2019).[2] Some of the most prominent contextual models are ELMo (Peters, et al., 2018), GPT (Radford et al., 2018), BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019). These are neural network language models which are usually trained on large corpus of text data using unsupervised objectives. In order to apply a pre-trained model to a specific language understanding task such as NLI, it needs to be fined-tuned by adding task-specific layers and training on task-specific labeled data. To this end, these pre-trained models can be considered as the encoder which provides fine-grained contextual embeddings for downstream models.

Despite the success of those strong pre-trained language models, it has been shown that they are limited by its lack of comprehension of the world.[3] They can be further improved by incorporating extra knowledge. A recent study suggested that the current NLU models suffer from insufficient contextual semantic representation and proposed to enrich sentences with predicate-specific argument sequences. By incorporating semantic role labels with BERT, the model for NLI has achieved an accuracy score of 91.9% on the SNLI dataset (Zhang et al., 2020).

We are particularly interested in improving NLI for Vietnamese. Due to the lack of annotated datasets, there has not existed much work on Vietnamese NLI. The most recent work about supervised Vietnamese NLI was published in 2015 (Nguyen et al., 2015) in which an experimental study of using the support vector machines (SVM) was conducted on a small dataset of 1600 sentences. These sentences are translated from the RTE-3 dataset (Giampiccolo et al., 2007). Another published work dates back in 2012, where a machine translation approach was proposed for Vietnamese NLI (Pham et al., 2012). The experiments were also conducted on a small translated dataset from RTE-3.

With the rise of applying pre-training methods in language processing, some recent works have been applied to Vietnamese language processing in recent years. Pre-trained language models, especially BERT-based models, have helped produce improvements for a variety of tasks. Bui et al. presented a study on using multilingual BERT embeddings and some new neural models for improving sequence

[2] As of September 15, 2020 on the latest GLUE test set.

[3] https://cs.nyu.edu/faculty/davise/papers/GPT3CompleteTests.html.

tagging tasks for Vietnamese, achieving new state-of-the-art results on part-of-speech tagging and named entity recognition (Bui et al., 2020). A little bit later, two pre-trained PhoBERT language models for Vietnamese were introduced where a specific word segmentation method was incorporated in order to take into account the difference between syllable-level and word-level representation. These pre-trained monolingual models improved four downstream tasks, including part-of-speech tagging, named entity recognition, dependency parsing and natural language inference (Nguyen & Nguyen, 2020). In particular, on the NLI task, these models are fined-tuned on a training set of 392,702 samples (which is released as a machine-translated version of the corresponding English training set of the XNLI project), the validation set and test set are manually-constructed, containing 2490 and 5010 samples respectively. They obtained an accuracy of 78.5% with the base model and of 80.0% with the large model respectively.

In this work, we focus on developing and experimenting with supervised models for NLI, including BERT without pre-training. To our knowledge, this present work is the first one that proposes neural network models for Vietnamese NLI. In addition, the experiments are conducted on the Vietnamese portion of the XNLI corpus, which is nearly five times larger than existing Vietnamese NLI corpus ever evaluated.

2.2 Integration of commonsense knowledge into deep learning models

The importance of background knowledge in natural language understanding has long been recognized. There has been a surge of interest in developing methods which allow integration of linguistic and commonsense knowledge in deep learning models. Earlier systems mostly exploited restricted linguistic knowledge such as manually-encoded morphological and syntactic patterns. With the advanced development of knowledge base construction, large amounts of semantic knowledge become available, ranging from manually annotated semantic networks like WordNet to semi-automatically or automatically constructed knowledge graphs like DBPedia (Lehmann et al., 2015) and NELL (Carlson et al., 2010). More recently, neural sequential models leverage the lower-dimensional real-valued representation of knowledge concepts as additional inputs such as KBLSTM (Yang & Mitchell, 2017). However, these models have treated the computation of neural sequential models as a black-box without tight integration of knowledge and computational structure. Most recently, Ma et al. (2018) proposed Sentic LSTM, an extension of the LSTM model which is capable of tightly integrate the commonsense knowledge of SenticNet (Cambria et al., 2020) into the recurrent encoder. This method is interesting in that it can exploit external knowledge to generate the hidden outputs and controlling the information flow, thereby outperform state-of-the-art methods in target-dependent aspect sentiment tasks.

Learning word representations for sentiment analysis has been an active topic of research recently. It has been shown that incorporating prior linguistic knowledge into deep learning models has the potential to learn better representations for sentiment analysis (Li et al., 2020; Peng et al., 2017). To this end, the newly introduced

AffectiveSpace 2 model, a general vector space model for concept-level sentiment analysis that allows for reasoning by analogy on natural language concepts, even when these are represented by highly dimensional semantic features (Cambria et al., 2015). This model can be regarded as a general framework for analogical reasoning that can be embedded in potentially any cognitive system dealing with real-world semantics, not only for NLP tasks but also for multimodal data processing (Poria et al., 2015). In addition, the study of microtext classification based on different methods, including phonetic based approaches has gained attraction, as presented in a recent comprehensive review (Satapathy et al., 2020).

Other work has focused on integrating knowledge bases into neural architectures for specific tasks such as reading comprehension (Wang & Jiang, 2019; Mihaylov & Frank, 2018), question answering (Sun et al., 2018; Bauer et al., 2018).

Transformer-based approaches have become a cornerstone in NLP systems, achieving state-of-the-art results in a wide variety of NLP tasks (Vaswani et al., 2017; Devlin et al., 2019). There have been some approaches that leverage knowledge bases like WordNet and DBPedia to fine-tune the internal hidden states of language models such as KnowBERT (Peters et al., 2019). Most recently, at SemEval2020, the Commonsense Validation and Explanation (ComVE) task was proposed to evaluate deep learning algorithms and models against commonsense tasks (Wang et al., 2020). The general purpose of the task is to test whether an NLP system can differentiate statements that make sense from those that do not. JUSTers, one participating system in that task has evaluated five-pretrained transformer-based language models, achieving good performance scores (Fadel et al., 2020).

In the last two years, there has been increasing interest in knowledge extraction and integration for deep learning architectures, which is the main topic of the DeeLIO workshop series (Deep Learning Inside Out). Lauscher et al. proposed lexically informed BERT (LIBERT) which integrates the discrete knowledge on word-level semantic similarity into pretraining (Lauscher et al., 2020). This research group have also investigated models for complementing the distributional knowledge of BERT with conceptual knowledge from ConceptNet using adapter training to improve BERT results (Lauscher et al., 2020). Also two recent approaches to infuse knowledge graphs into pretrained language models to improve social commonsense task have been proposed (Chang et al., 2020). These methods are evaluated on the SocialIQA dataset.

3 Methods

In our approach, each sentence, either a premise or a hypothesis, is encoded by a real-valued dense vector by a computational model. In this work, we propose four different model architectures which are employed and compared. These architectures make use of different neural models, including bag-of-words (BOW) model, convolutional neural network (CNN), gated recurrent unit (GRU) network and the transformer. Given an input sentence of n tokens, $s = [w_1, w_2, \dots, w_n]$, each token

$w_t, t = 1, \ldots, n$ is represented by an embedding vector of u dimensions. Each model computes an output vector of v dimensions for the entire input sentence s.

The following subsections present in detail the four architectures. They are purely neural network based models. The integration of semantic information into these models will be described in the next section.

3.1 Bag-of-words model

BOW is the most simple model in which each sentence is represented as the sum of the embedding representations of its words. More precisely, given a premise sentence of m words $[p_1, p_2, \ldots, p_m]$, each word p_t is embedded into a vector $e(p_t) \in \mathbb{R}^u$ and the BOW encoder simply computes the premise embedding as $e(p) = \sum_{t=1}^{m} e(p_t)$. In this model, the premise dimension is always the same as that of token embeddings, that is $v = u$. Similarly, the embedding of each hypothesis sentence of n words $[h_1, h_2, \ldots, h_n]$ is computed as $e(h) = \sum_{t=1}^{n} e(h_t)$.

Two sentence embeddings $e(p)$ and $e(h)$ are then concatenated to get a vector of $2u$ dimensions, which is passed to a single tanh layer, followed by a linear and a three-way softmax classifier. Given a premise and hypothesis pair, the model produces a probability distribution of that pair bearing an entailment (E), a neutral (N) or a contradiction (C) relationship. Figure 1 illustrates the BOW model architecture.

3.2 Sequential model

In this architecture, the premise and hypothesis sentences are processed sequentially in order. The rationale for this is that the premise could establish the context for the hypothesis. The input to the sequential model is the sequence of $m + n$ words $[p_1, p_2, \ldots, p_m, h_1, h_2, \ldots, h_n]$. This word sequence is encoded by an encoder of type CNN, uni-directional GRU or bi-directional GRU. With the CNN encoder, we use a $1d$ convolutional layer with 5 filters and the rectifier linear activation function, followed by a global max pooling layer to extract salient hidden features. With GRU encoders, we can view that the final hidden state of the premise becomes the initial hidden state for the hypothesis GRU, as depicted in the Fig. 2.

In this work, we choose GRU as the recurrent unit rather than the Long Short-Term Memory (LSTM) unit. The GRU is like LSTM with forget gate but has fewer parameters than LSTMs, as it lacks an output gate (Kyunghyun et al., 2014). Through many experiments, we see that GRUs give better results than LSTMs while being faster to train. GRUs have been shown to exhibit even better performance on certain smaller datasets (Chung et al., 2014).

A GRU has two gates, a reset gate r and an update gate z. Intuitively, the reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. If we set the reset gate to all one and update gate to all zero, we get the plain recurrent model. The equations of the GRU unit at each time step t are as follows:

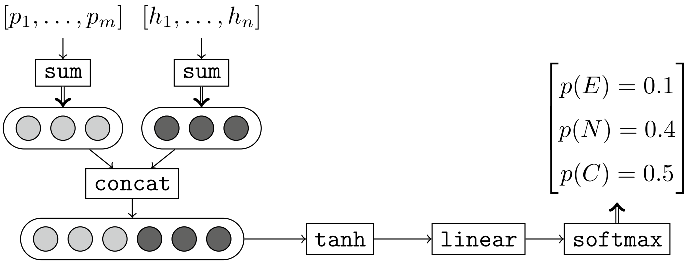


Fig. 1 The continuous BOW model

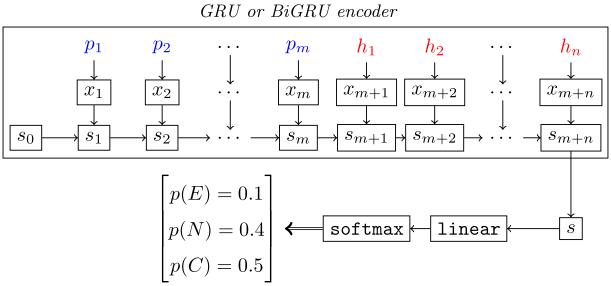


Fig. 2 Sequential model architecture with RNN encoder. $x_t \in \mathbb{R}^d$ are token embeddings and $s_t \in \mathbb{R}^o$ are hidden states of a GRU at time step t. The last output state is taken as the embedding vector of both the premise and hypothesis

$$
\begin{aligned}
z_t &= \sigma\left(W^z x_t + U^z s_{t-1} + b^z\right)\\
r_t &= \sigma\left(W^r x_t + U^r s_{t-1} + b^r\right)\\
u_t &= \tanh\left(W^u x_t + U^u (s_{t-1} \odot r_t) + b^u\right)\\
s_t &= (1 - z_t) \odot u_t + z_t \odot s_{t-1},
\end{aligned}
$$

where $W^{\cdot}, U^{\cdot}$ are parameter matrices and $b^{\cdot}$ are bias vectors, and $\sigma(\cdot)$ is the sigmoid function. In this work, we use both the unidirectional and bidirectional GRU model to allow capturing both past and future information at each sequence position. This model consists of two GRUs which are run in parallel, one on the input sequence and the other on the reverse of the input sequence. At each time step, the hidden state of the bidirectional model is the concatenation of the forward and backward hidden states are concatenated, that is $x_t = \overrightarrow{x_t} \oplus \overleftarrow{x_t}$.

The recurrent network block in the model can also be replaced by a CNN. The network learns filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature engineering is a major advantage of CNN.

We build our CNN upon that of Kim (2014) which is originally proposed for sentence classification. The CNN block consists of a $1d$ convolutional layer to recognize w-grams, a non-linear layer with the rectifier activation function, and a max pooling layer to extract the most relevant features.

The input sequence can be viewed as a tensor of token embeddings $X = [x_1, x_2, \dots, x_{m+n}]^\top$ of size $(m+n) \times d$. This matrix is fed into the convolutional layer to extract higher level features. Given a window size w, a filter is seen as a weight tensor F of size $o \times d \times w$, where o is the output frame size of the filter. The core of this layer is obtained from the application of the convolutional operator on the two tensors X and F. The output layer of the convolutional layer is precisely computed as

$$Y_{ti} = \sum_{j=1}^{d} \sum_{k=1}^{w} F_{ijk} * X_{t-1+k,j} + b_i,$$

for all $t = 1, 2, \dots, n - w + 1, \forall i = 1, 2, \dots, o$, where $b = [b_1, b_2, \dots, b_o]$ is the bias tensor of size o. Then a rectifier linear unit layer is applied element-wise on the output layer to produce score tensor.

The pooling is then applied to further aggregate the features generated from the previous layer. The popular aggregating function is max as it bears responsibility for identifying the most important features. More precisely, the max pooling layer produces $z = [z_1, z_2, \dots, z_o]$, where $z_i = \max_{1 \le t \le n-w+1} Y_{ti}$.

3.3 Parallel model

In the parallel architecture, the premise and hypothesis sentences are encoded separately. The rationale for this encoding scheme is that the model might have a chance to find rich abstract relationships between them. In addition, these sentences encoding could facilitate transfer to other tasks. As in the sequential architecture, each sentence can be encoded by using a CNN, a uni-directional GRU or bi-directional GRU.

Figure 3 depicts the parallel model. After concatenating the last output states of the premise and hypothesis encoders, the resulting $2o$-dimensional vector representation is passed to a linear layer and then a three-way softmax layer to compute the predictive distribution.

3.4 BERT model

Many NLP tasks have been shown to greatly benefit from large network pre-trained models. In recent years, these pre-trained models have led to a series of breakthroughs in language representation learning (Radford et al., 2018; Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019; Clark et al., 2020). Introduced in late 2018, BERT (Devlin et al., 2019) stands for Bidirectional Encoder Representation from Transformers, which is designed to pre-train deep bidirectional language representations by jointly training on both left and right contexts of a given word in all layers of the model. The core of BERT's model architecture is a multi-layer bidirectional transformer encoder, which was proposed by Vaswani et al. (2017). Transformers dispense entirely with recurrence and convolution mechanisms and rely solely on *attention mechanisms*, which significantly decreases training time.

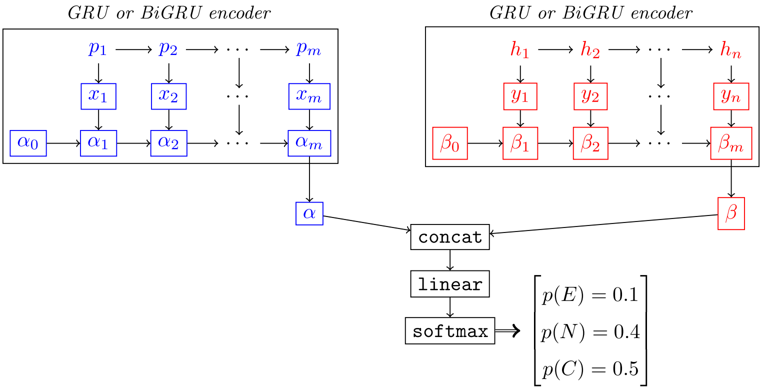


Fig. 3 Parallel model architecture with RNN encoders. $x_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}^d$ are token embeddings of the premise and hypothesis respectively. The last output states α and β of the two GRUs are concatenated as the joint embedding of the premise and hypothesis pair

In this work, we implement two BERT model architectures for natural language inference and compare them with other models. One model is purely supervised which is trained from scratch on the same training set. One model employs a state-of-the-art pre-trained language model for Vietnamese (PhoBERT-base) which is fine-tuned on the training set. For completeness, we describe briefly this model as follows.

BERT relies on several layers of transformers blocks, as shown in Fig. 4, where Trm are transformers and E_k are embeddings of the k-th token.

Each transformer block consists of two sub-layers, a multi-head self-attention mechanism followed by a simple position-wise fully-connected feed-forward network. Residual connections exist around each of the two sub-layers, and dropout, following after each sub-layer, provides layer normalization, as shown in Fig. 5.

In essence, the multi-head attention layer in the transformer architecture encodes a value V according to the attention weights from query Q to key K. If $\mathcal{G}_f$ is a position-wise feed-forward network, then the transformer $\mathcal{F}(Q, K, V)$ computes an output as follows

$$O = V + \mathcal{G}_f(\texttt{MultiHead}(Q, K, V)),$$

where Q, K, V, O are matrices of size $N \times o$. The attention mechanism is performed in parallel for each token in the sentence to obtain their updated features in one shot. This parallel computation offers a plus point for transformers over recurrent network models. Given the concatenated token embeddings E of premise and hypothesis sentences, the transformer $\mathcal{F}(E, E, E)$ is then used to encode E and the last hidden state of the output is then used to perform classification as in the recurrent models.

The pre-trained PhoBERT model used in this work is the base version, using the same architecture of BERT$_{\text{base}}$ (Nguyen & Nguyen, 2020). Its pre-training approach is based on RoBERTa (Liu et al., 2019) which optimizes the BERT pre-training

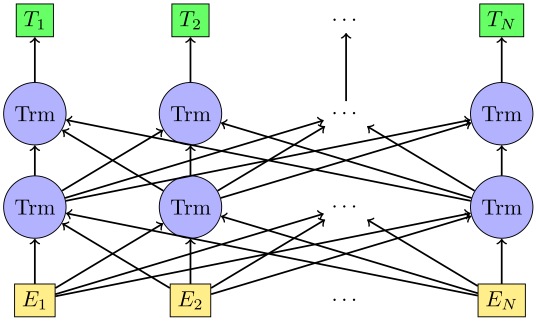
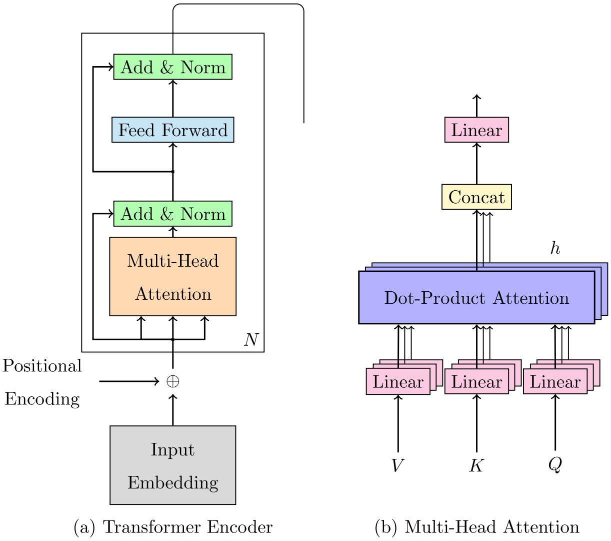


Fig. 4 BERT architecture



(a) Transformer Encoder

(b) Multi-Head Attention

Fig. 5 Transformer architecture (Vaswani et al., 2017)

procedure for more robust performance. The model was trained on a 20GB dataset of uncompressed texts.

4 Integration of semantic information

This section presents our method for incorporating lexical and concept-level semantic information into the machine learning models to improve their performance. The integration process has three main stages, including word segmentation, semantic lookup, and incorporation of semantic concepts into the proposed models.

4.1 Word segmentation

The first important task is to segment a sentence, either a premise or a hypothesis into lexical units. Many languages use alphabetic script, which usually separate words by blanks and a tokenizer which simply replaces blanks with word boundaries and cuts off punctuation marks, parentheses and quotation marks at both ends of a word, is already quite accurate. However, many languages, including Vietnamese, blanks are not only used to separate words, but they are also used to separate syllables that make up words. Furthermore, many syllables are words by themselves, but can also be part of multi-syllable words whose syllables are separated by blanks between them. This phenomenon creates problems for all NLP tasks, complicating the identification of what constitutes a word in an input text.

The dataset which is used in this study contains Vietnamese sentences. For example, two sample sentences in the dataset are as follow:

- *Anh ấy rất trung thành và tử tế.* (*He is very faithful and nice.*)
- *Tôi ghét anh ta vì anh ta quá kiêu ngạo.* (*I hate him because he is too arrogant.*)

Each sentence can be considered as a list of syllables, for example, the list of syllables of the first example sentence is

[“*Anh*”, “*ấy*”, “*rất*”, “*trung*”, “*thành*”, “*và*”, “*tử*”, “*tế*”, “.”]

However, the lexical units, or words, the smallest units which have meaning in that sentence are as follows:

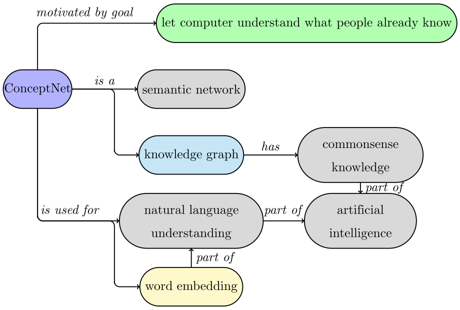
[“*Anh*”, “*ấy*”, “*rất*”, “*trung_thành*”, “*và*”, “*tử_tế*”, “.”]

In this example, in addition to monosyllabic words, there are two disyllabic words which should be segmented correctly—“*trung_thành*” (*faithful*) and “*tử_tế*” (*nice*). Word segmentation of Vietnamese text is itself an interesting problem. In this work, we perform word segmentation using the hybrid method proposed by Le-Hong et al. (2008), which is both fast and accurate. Naturally, we can either consider a syllable or a word as a token and feed them into the proposed models. However, for semantic lookup and integration, only word-level tokens make senses. For this reason, semantic-aware models is concerned with only word-level tokens.

4.2 Part-of-speech tagging

The second pre-processing step that we perform is part-of-speech tagging. Because of its inflectionless nature, Vietnamese does not have morphological aspects such as gender, number, case...such as in occidental languages. Vietnamese words are classified based on their combination ability, their syntactic functions and their general meaning. Beyond the classical part-of-speech tags which are used in Western languages (noun, verb,...), the Vietnamese tagset has classifiers, which are commonly found in Asian languages, and modal words, which convey some of the nuances borne by flection in synthetic languages (Le-Hong et al., 2010).

Fig. 6 An illustration of ConceptNet in graph



After word segmentation, we use a statistical part-of-speech tagger to label word sequences of both premise and hypothesis sentences with their part-of-speech tags. Important words are then filtered based on their word category. A word is considered important if it is a noun, either proper noun (of tag *Np*), unit noun (*Nu*) or common noun (*N*), a pronoun (*P*), a verb (*V*), an adjective (*A*). The important word category set includes five tags out of 18 part-of-speech tags defined by the VLSP Vietnamese treebank.[4]

4.3 Semantic lookup

In this work, we propose to include lexical semantic information provided by ConceptNet into the deep learning models. ConceptNet (Speer et al., 2017) is a freely-available semantic network, designed to help computers understand the meanings of words that people use. ConceptNet originated from the crowdsourcing project Open Mind Common Sense, which was launched in 1999 at the MIT Media Lab. It has since grown to include knowledge from other crowdsourced resources, expert-created resources, and games with a purpose.[5]

Figure 6 shows an illustration of ConceptNet in graph. Much of ConceptNet knowledge comes from Wiktionary, the free multilingual dictionary. This gives us information about synonyms, antonyms, translations of concepts into hundreds of languages, and multiple labeled word senses for many words. ConceptNet also connects to a subset of DBPedia,[6] which extracts knowledge from the infoboxes on Wikipedia articles. In addition, ConceptNet integrates dictionary-style knowledge that comes from Open Multilingual WordNet,[7] which provides access to open WordNets in a variety of languages, all linked to the Princeton WordNet of English (PWN).[8]

The nodes of ConceptNet are words and phrases of natural language. Each node has a URI within ConceptNet that starts with `/c/` and a language code, such as `/c/en/beautiful`. Given a word or phrase, we will know the complete URL, then we can look it up using a publicly available API. The actually interesting information is inside the `edges` list that connect a node with other nodes. Each edge specifies a relation between a start node and an end node. In the current version of

[4] http://vlsp.org.vn/.

[5] http://conceptnet.io/.

[6] https://wiki.dbpedia.org/.

[7] http://compling.hss.ntu.edu.sg/omw/.

[8] http://wordnet.princeton.edu/.

ConceptNet, a set of 34 relations are defined that can apply to text in any language. The relations are given canonical, camel-cased English names in the `/r/` namespace, such as `/r/PartOf`. Table 2 lists 12 relations along with their description and examples. For a full list of relations defined in ConceptNet, the reader may refer to its website.[9] These relations are used in this study to extract semantic information from text.

As an example, consider the word "*vào*" in Vietnamese. This word has multiple senses and its correct meaning depends on the context that it appears in a given sentence. Using ConceptNet, we can look up all the words and phrases that have an interesting relation with it, as specified by the relations defined in Table 2. More interestingly, these relations are specified in across different languages such as English (*en*), French (*fr*), German (*de*), or Vietnamese (*vi*).

As shown in Fig. 7, ConceptNet provides us the synonyms of the source word "*vào*" both in Vietnamese (*vô*), and in English (*enter*), and also in German (*herein*). Antonyms are also important for detecting contradiction relation and many lexical units in ConceptNet has references to their antonyms. The Vietnamese antonym of "*vào*" is "*ra*" (go out/out). In addition, we can pull out related concepts of that word in these languages, and French if there is any.

4.4 Semantic integration

We propose a method to enrich both premise and hypothesis sentences with concept-level semantics information. For each premise and hypothesis pair p and h, the method has the following steps:

1. Tokenize p and h into words;
2. Tag the word-based sequences with their parts-of-speech;
3. Filter important words by using parts-of-speech of types pronoun, noun, verb and adjective;
4. Enrich the filtered words with their end nodes information in the ConceptNet by looking them up using a pre-defined set of semantic relations. We now have a bag of concepts for premise sentence p and a bag of concepts for the corresponding hypothesis sentence h.
5. Encode the bag of concepts with the BOW-like model and integrate resulting real-valued vectors into the proposed models.

Figure 8 illustrates the steps. For illustration, we take three samples from the Vietnamese XNLI dataset where the premise sentence is "*Tên của tiền cũng được lấy từ các thứ và các loài động_vật.*" ("*Money has also derived its names from things or animals.*"). Three hypothesis sentences with the corresponding inference label are as follows:

[9] https://github.com/commonsense/conceptnet5/wiki/Relations.

Table 2 Some relations defined in ConceptNet version 5 which are used in our work

No	Relation URI	Description	Examples
1	/r/IsA	A is a subtype or a specific instance of B; every A is a B. This is the hyponym relation in WordNet	car → vehicle; Chicago → city
2	/r/PartOf	A is a part of B. This is the part meronym relation in WordNet	gearshift → car
3	/r/HasA	B belongs to A, either as an inherent part or due to a social construct of possession. HasA is often the reverse of PartOf	bird → wing; pen → ink
4	/r/UsedFor	A is used for B; the purpose of A is B	bridge → cross water
5	/r/CapableOf	Something that A can typically do is B	knife → cut
6	/r/MadeOf	A is made of B	bottle → plastic
7	/r/DefinedAs	A and B overlap considerably in meaning, and B is a more explanatory version of A	peace → absence of war
8	/r/Causes	A and B are events, and it is typical for A to cause B	exercise → sweat
9	/r/Desires	A is a conscious entity that typically wants B. Many assertions of this type use the appropriate language's word for "person" as A	person → love
10	/r/Synonym	A and B have very similar meanings. This is the synonym relation in WordNet as well. Symmetric	sunlight ↔ sunshine
11	/r/Antonym	A and B are opposites in some relevant way, such as being opposite ends of a scale, or fundamentally similar things with a key difference between them. This is the antonym relation in WordNet as well. Symmetric	black ↔ white; hot ↔ cold
12	/r/RelatedTo	The most general relation. Symmetric	learn ↔ erudition

The numbers in bold are the best score for each column

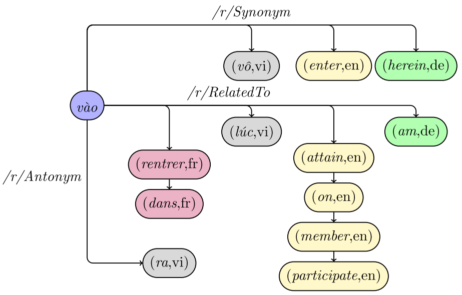


Fig. 7 Cross-language relations of the Vietnamese word "*vào*"

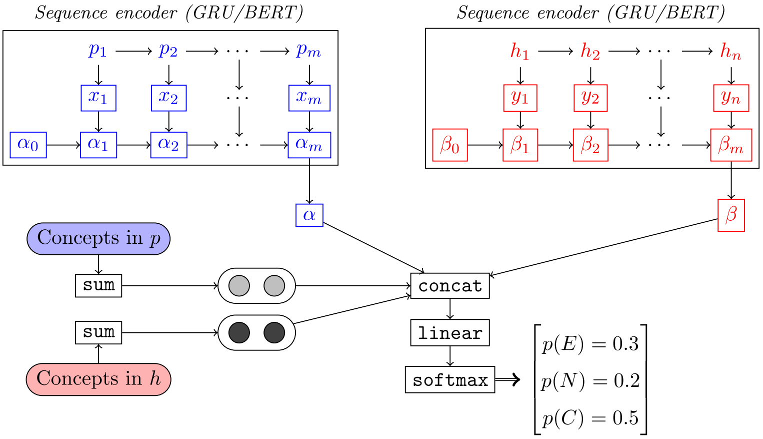


Fig. 8 A semantic-enriched parallel model architecture. $x_t \in \mathbb{R}^d$ are token embeddings and $s_t \in \mathbb{R}^o$ are hidden states of a GRU at time step t. The last output states α and β of the two GRUs are concatenated as the joint embedding of the premise and hypothesis pair

- entailment: "*Tên của tiền được lấy từ động_vật.*" (*Money got its name from animals.*)
- contradiction: "*Tiền không được đặt tên theo động_vật.*" (*Money was not named after animals.*)
- neutral: "*Một đồng xu được đặt theo tên một con sư_tử.*" (*One coin is named after a lion.*)

The content words of noun and verb category in the premise are "*tên*", "*tiền*", "*lấy*", "*thứ*", "*loài*", and "*động_vật*". These important word also appear in the first and the second hypothesis. The third hypothesis has three more content words which are "*đồng*", "*xu*" and "*sư_tử*". Looking up each of these content words in the ConceptNet with the pre-defined semantic relations, we can extract the following concepts:

- "*tên*":
  - Vietnamese: `/r/Synonym` → {mũi_tên}, `/r/RelatedTo` → {họ, cung, nỏ}

  - English: /r/RelatedTo → {arrow}
  - French: /r/RelatedTo → {flèche, placé, devant, mauvais}

- "*tiền*":
  - Vietnamese: /r/IsA → {tiền}
  - English: /r/Synonym → {money}, /r/RelatedTo → {pre}
  - French: /r/RelatedTo → {argent}
  - German: /r/Synonym → {geld}

- "*lấy*":
  - English: /r/RelatedTo → {steal, charge, marry, wed, take, seize }
  - French: /r/RelatedTo → {soi_même, extraire, retirer, voler, fonder, dérober, embaucher, ôter, lever, emparer, moyens, enlever}

- "*thứ*":
  - English: /r/RelatedTo → {object, inferior, sort, rank, vice, second, pardon, type, kind, order, under, forgive}
  - French: /r/RelatedTo → {objet, rang, jour, pardonner, second, passable, sorte, chose, espèce}

- "*loài*":
  - English: /r/RelatedTo → {species}
  - French: /r/RelatedTo → {espèce, catégorie}

- "*động_vật*"
  - Vietnamese: /r/IsA → {động_vật}
  - English: /r/RelatedTo → {animal}
  - French: /r/RelatedTo → {animal}

- "*đồng*":
  - English: /r/RelatedTo → {currency, dong, medium, thousand, sorcerer, field}
  - French: /r/Related → {champ}

- "*xu*":
  - English: /r/RelatedTo → {penny, xu, cent}

- "*sư_tử*":
  - Vietnamese: /r/IsA → {động_vật}
  - English: /r/RelatedTo → {lion, leo}
  - French: /r/RelatedTo → {lion}
  - German: /r/Synonym → {löwe}

It can be seen that these concepts represent rich semantic information which may help models in inference. For example, they have cross-lingual information that lion is an animal or first name is related to a family name. In our method, all the 12 semantic relations presented in Table 2 are used. For the multilingual relation links, we incorporate all available links from all languages for which the data are available. For Vietnamese concepts, we observe that most links point to popular languages such as English, French or German.

It is worth noting that, in the example above, we are concerned with Vietnamese, which is an isolating language. Vietnamese is a monosyllabic language and its word forms never change, contrary to occidental languages that make use of morphological variations (plural form, conjugation, etc.). For this reason, we can lookup concepts in the Vietnamese concept net directly, without any text pre-processing other than word segmentation. However, when experimenting with English, we need an extra pre-processing step called stemming to reduce inflected or derived words to their word stem before word lookup. For example, the word "*cats*" should be reduced to "*cat*" before searching for this concept with ConceptNet.

5 Experiments

5.1 Datasets

We test our proposed method on two natural languages, including Vietnamese and English. The two datasets are drawn from the XNLI corpus.

The Vietnamese portion of the XNLI corpus contains 7500 sentence pairs. We randomly split this corpus into a training set and a test set with the ratios of 80% and 20% respectively. The average number of tokens per sentence in the XNLI corpus for Vietnamese is 27.6 for premises and 13.5 for hypotheses. The English portion of the same corpus also contains 7500 sentences. This corpus is also split into a training set and a test set with the same ratio 80/20.

5.2 Training details

All models are initialized with random token vectors rather than reference pre-trained word vectors. This allows to establish results in a pure supervised learning setting rather than a semi-supervised or transfer learning setting. The models are all trained by the Adam optimizer (Kingma & Ba, 2015) with default parameters. We use the cross-entropy loss function for 3-way classification as usual. All models are trained in 50 epochs.

We first evaluate the performance of the models with respect to the token embedding size, which varies in the set $\{25, 50, 80, 100\}$. This experiment allows investigation of the effect of input embedding size on the performance. In the experiments with the sequential and parallel models, we set the number of hidden units of the CNN or GRU models to different values ranging from 25 to 300. These varied sizes allow us to evaluate the effect of encoder size to the accuracy of the proposed models. Each model is trained repeatedly five times with different randomly initialized parameters, and their scores are averaged.

In the experiments with the BERT models, we vary the number of transformer blocks (or layers) in the set $\{1, 2, 4, 8\}$, the hidden size ranges from 8 to 304[10] while keeping the number of self-attention heads fixed at 8 and the intermediate size (i.e, feed-forward) fixed at 256. These varied parameters allow us to evaluate the scalability of the models, i.e. how their performance changes as they are made to have more parameters or layers. As in sequential and parallel models, each BERT model is trained repeatedly five times with different randomly initialized parameters, and their averaged scores are reported. We observed that training a BERT model requires significantly more time than that of other models.

All the models presented in this paper, except the models using pre-trained PhoBERT, are implemented by ourselves in the Scala programming language. We use the BigDL and Analytic Zoo libraries[11] as the deep learning framework. These libraries provide an end-to-end pipeline for applying AI models and high-level machine learning workflow for automating machine learning tasks. Furthermore, we can quickly code inline with Apache Spark[12] code for distributed training and inference, easily scale out on multiple commodity servers, without relying on expensive GPU devices. The pre-trained models utilizing PhoBERT are implemented using PyTorch, an optimized tensor library for deep learning using GPUs and CPUs.[13] Specifically, we load the *vinai/phobert-base* model with 12 pre-trained transformer blocks, producing a context vector of 768 dimensions for each sample, and then feed this vector to a RoBERTa classification head to perform 3-way prediction. Our code and detailed experimental results are publicly available in a GitHub directory.[14]

5.3 Experimental results

In this subsection, we present the result of four sets of experiments. The first three concern Vietnamese NLI and the fourth one concerns English NLI. The first experiment set compares the performance of four neural network models: continuous bag-of-words model, sequential model, parallel model, and BERT model as presented in Sect. 3 using the syllable-based inputs. The second experiment set is similar to the first one except that word-based inputs are used rather than syllable-based inputs. The third experiment set presents results of semantics-enriched models as described in Sect. 4. Finally, the fourth set presents results for English NLI.

5.3.1 Syllable-based performance

The training and test accuracy of the BOW model are shown in Table 3. It seems that for this model, increasing the token embedding size makes the model overfit—it

[10] Note that in transformers-based models, the hidden size must be a multiple of the number of self attention heads.

[11] https://analytics-zoo.github.io/.

[12] http://spark.apache.org.

[13] https://pytorch.org.

[14] https://github.com/phuonglh/vlp/, under the *nli* module.

Table 3 Performance of the BOW model

Token Embedding Size	Train. Score	Test Score
25	0.7665	**0.3641**
50	0.7878	0.3455
80	0.7878	0.3438
100	0.7878	0.3347

The numbers in bold are the best score for each column

The best score is 36.41% with a small token embedding size of 25

Table 4 Performance of the sequential and parallel model using CNN encoder or GRU encoder with respect to syllable embedding size *d* and encoder output size *o* on the test set

	Sequential Model				Parallel Model			
	Token Embedding Size (*d*)				Token Embedding Size (*d*)			
	CNN Encoder		GRU Encoder		CNN Encoder		GRU Encoder	
Size (*o*)	25	50	25	50	25	50	25	50
100	0.2703	0.2599	0.4089	0.4512	0.4013	0.4030	0.4276	0.4168
128	0.2620	0.2680	0.4253	0.3920	0.3911	0.4103	0.4634	0.4388
150	0.2685	0.2683	0.5012	0.4163	0.4145	0.4089	0.4221	**0.5018**
200	0.2637	**0.2709**	0.5057	0.4942	0.4207	0.4443	0.4566	0.4965
256	**0.2784**	0.2643	0.4999	**0.5057**	0.4336	0.4151	0.5008	0.4788
300	0.2746	0.2759	0.4999	0.4938	0.3992	**0.4449**	0.4954	0.4930

The numbers in bold are the best score for each column

gives better training accuracy, up to a limit of its generalization capability, but has worse test accuracy.

In the sequential model, the GRU encoder outperforms the CNN model. The gap is quite large, of about 22.7% of F_1 score on average. More precisely, the best sequential model with the CNN encoder has a test score of 27.84% with the encoder size of 256 and the token embedding size of 25; this score is even worse than a random guess for 3-way classification. Meanwhile, with the GRU encoder, the sequential model attains its best score of 50.57%.

The CNN encoder performs better in the parallel model than that in the sequential one. It attains an accuracy of 44.49%, as shown in Table 4. The GRU-encoder parallel model has its peak score of 50.18%. Since the size of the dataset is relatively small, we see in the experiments that a small value of token embedding size of 25 or 50 gives better results than large token embedding sizes. We observe that the GRU encoder performs quite similarly in both the sequential and parallel model.

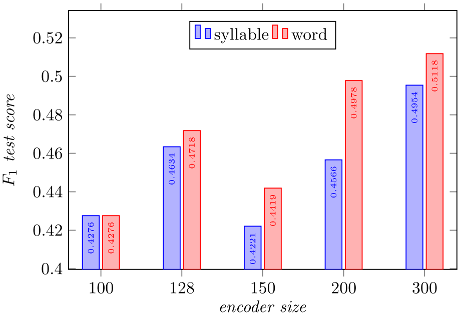
The BERT models outperform both sequential and parallel ones, as shown in Table 5. On this relatively small corpus, using a minimal number of transformer

Table 5 Performance of the BERT models with respect to the number of transformer blocks on the test set. Here, the number of self-attention heads is fixed at 8 and the intermediate size is fixed at 256

Blocks	Best Test F_1	Best Encoder Size
1	**0.5275**	200
2	0.5109	128
4	0.5036	80
8	0.4952	64

The numbers in bold are the best score for each column

Fig. 9 Word-based parallel models versus syllable-based parallel models with the GRU encoder



blocks (or layers) gives better scores than using a large number of blocks. The best test F_1 score is 52.75% with one self-attention head and 200 hidden units.[15]

5.3.2 Word-based performance

We next compare the performance of the models on word-segmented input sentences. The input sentences are first split automatically in lexical units before feeding into the word embedding layer of the proposed models, as described in the Sect. 4.1. As in the syllable-based experiments, we conducted a series of experiments with different models. We observed the same fact that the BOW models are all outperformed by the others. For brevity, in this subsection, we report the experimental results of better models, including the parallel models with GRU encoder and the BERT models.

As shown in Fig. 9, with parallel architecture, the word-based models are more accurate than its syllable-based counterpart by an average test score of 1.36% of absolute points. The word-based models achieve the best score of 51.18% with the GRU encoder of 300 hidden units.

Figure 10 shows the comparison between the syllable-based models, the word-based models and semantics-enhanced models when using the BERT architecture with different numbers of transformer blocks. The word-based model achieves its best score of 53.65%, about 0.9% of absolute score better than the best syllable-based supervised BERT model.

[15] More detailed experimental results can be found in our GitHub repository.

Table 6 Performance of the sequential models with semantic information integrated

Size	CNN Encoder		GRU Encoder		Δ CNN	Δ GRU
	25	50	25	50		
100	0.3317	0.3540	0.4731	0.4840	*0.0837*	*0.0328*
128	0.3528	0.3464	0.4840	0.4875	*0.0848*	*0.0622*
150	0.3372	**0.3587**	0.4835	0.4805	*0.0902*	*– 0.0178*
200	0.3456	0.3549	0.4850	0.4997	*0.0840*	*– 0.0060*
256	0.3446	0.3402	0.4941	**0.5197**	*0.0662*	*0.0140*
300	0.3499	0.3507	0.4906	0.4972	*0.0748*	*0.0027*

The numbers in bold are the best score for each column

We test the statistical significance of the results by performing a paired sample t-test with $\alpha = 5\%$. The test result confirms that the differences between scores of the models are statistically significant, where the two-sided p-value is less than 10^{-7}.[16]

How do these purely supervised models compare to a pre-trained PhoBERT model? The PhoBERT base model fine-tuned on the same training set with the same input words achieves a test accuracy of 66.60%. This is quite a large gain compared to the best supervised word-based BERT model above. This experimental result demonstrates the importance of pre-trained language models for Vietnamese such as ViBERT (Bui et al., 2020) and PhoBERT (Nguyen & Nguyen, 2020) in improving Vietnamese language inference. Note that, this result is not comparable to the work presented in Nguyen and Nguyen (2020) since the training and test sets are not the same. As discussed in Sect. 2, the original PhoBERT model was fined-tuned on a training set of 392,702 samples and tested on 5,010 samples to obtain an accuracy of 78.5%.

5.3.3 Semantics-enriched performance

Table 6 presents the performance of the sequential model with CNN encoder or GRU encoder when semantic information is integrated. The last two columns of this table show the score gain in comparison to the best scores of the corresponding sequential model without semantic integration.

We observe a large gain on the CNN encoder. The best performing CNN-sequential model without semantic enhancement is 27.84%; while with semantic integration, this score is 35.87%, that is about 8% of absolute point. However, the performance gain with the stronger GRU-sequential model is smaller, of about 1.4%, from 50.57 to 51.97%. This peak score is also better than 51.18%, the best score of the GRU-parallel model without semantics integration reported in the previous experiment set.

[16] We use the package HypothesisTests of the Julia programming language to perform the statistical tests.

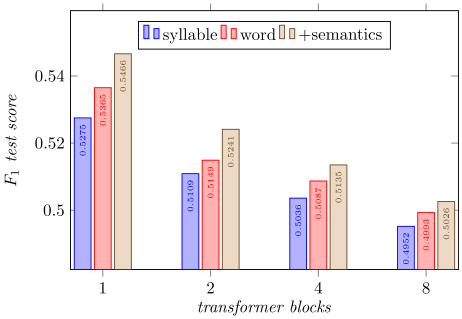


Fig. 10 Word-based models versus syllable-based BERT models using different number of transformer blocks

As shown in Fig. 10, with lexical semantics integration, the scores are pushed further by about 1% in average. The pre-trained PhoBERT model is also benefited by lexical semantics by about 1%. This result confirms effectiveness of using lexical units and lexical semantics integration into BERT architectures in language inference.

5.4 Performance on the English dataset

In the final set of experiments, we evaluate the proposed models on the English dataset. Since the BOW model is consistently outperformed by stronger ones and for saving space, we report only results of strong models including sequential, parallel and BERT models, without or with semantic information integration.

The performance scores of the sequential and parallel models are shown in the Table below.

Performance of the sequential and parallel model on the English dataset

Size	Sequential Model				Parallel Model			
	CNN Encoder		GRU Encoder		CNN Encoder		GRU Encoder	
	25	50	25	50	25	50	25	50
100	0.3235	0.3327	0.3844	0.4529	0.4102	0.4514	0.4307	0.4207
128	0.3249	0.3412	0.4841	0.4448	0.4164	0.4351	0.4482	0.4738
150	0.3301	0.3359	0.4587	0.4926	0.4176	0.4734	0.4356	0.4811
200	0.3225	0.3580	**0.4997**	0.4841	0.4570	0.4310	0.4803	0.4365
256	0.3369	0.3333	0.4841	0.4962	0.4057	0.4430	0.4911	**0.5119**
300	0.3416	0.3493	0.4876	0.4997	0.5137	0.4962	0.5032	0.4806

We see that the GRU encoder outperforms the CNN encoder in both sequential and parallel architecture. The best performance of the sequential model is 49.97% with 25-dimensional token embeddings and 200-dimensional encoder size. Meanwhile, the best performance of the parallel model is 51.19% with 50-dimensional token embeddings and 256-dimensional encoder size.

Figure 11 shows the test score of the BERT-based model on the English dataset with different encoder sizes. In this experiment, we use either one transformer block or two transformer blocks with 8 attention heads and 256 hidden states for the

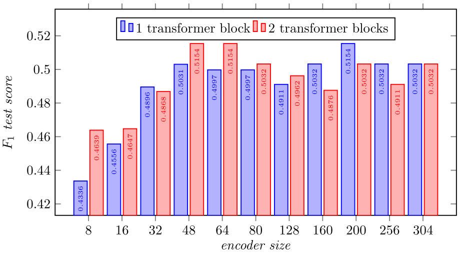


Fig. 11 F_1 scores of the BERT-based model on the English test set with different encoder sizes and numbers of transformer blocks

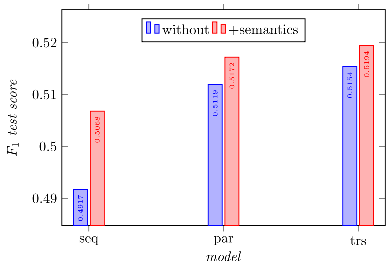


Fig. 12 Performance gain when semantic integration is integrated into the proposed model on the English dataset

intermediate size. We see that the peak performance of the model on the test set is about 51.54% when using 64 dimensions and two blocks or using 200 dimensions and one block.

Figure 12 presents the performance gain when semantic information is integrated into the proposed models. The sequential, parallel and BERT models improve by about 0.71%, 0.53% and 0.4% respectively when semantic information is integrated. On average, semantic integration improves the performance by about 0.54%. The BERT model gives the best score of 51.94% on the test set.

5.5 Discussion

The experimental results have shown that Vietnamese inference is difficult. The Vietnamese portion of the XNLI corpus, as well as nine other languages, are translated from the English portion by human translators. This translation approach carries with it the risk that the semantic relations between the two sentences in each pair might not be reliably preserved. Indeed, by investigating closely the Vietnamese dataset, we find that issue—many examples are not well written due to mediocre translation or semantic shift. This problem concerns many sentences. We show in Table 7 some of sentence pairs which highlight the problem.

A native Vietnamese speaker can easily see that nine samples above are results of a bad translation which makes the annotated labels incorrect or inconsistent. For example, the first three premise sentences mean *I did not have time to participate in all.* which are very different semantically to the original English sentence which says *I didn't have time to enter in all kinds of whatever.*. For this reason, the second and third hypothesis sentence which mean *I entered there in time.* and *I ran out of*

Table 7 Some examples which illustrate the semantic shift due to mediocre translation from English to Vietnamese

No	Premise	Label	Hypothesis
1	Tôi không có thời gian để tham gia tất cả. (*I didn't have time to participate in all.*)	neutral	Tôi lẽ ra có thể hoàn thành cập nhật nó sau đó. (*I could have been completing the update.*)
2	Tôi không có thời gian để tham gia tất cả	contra	Tôi đã vào đó đúng giờ. (*I entered there in time.*)
3	Tôi không có thời gian để tham gia tất cả	entail	Tôi hết thời gian để nhập tất cả vào. (*I ran out of time to enter it all in.*)
4	Và thực tế là cô ấy thật nhẹ nhàng! (*And the fact is she was light!*	neutral	Cô ấy ăn rất nhiều đồ ăn, nhưng vẫn giữ được trọng lượng của mình. (*She eats so much but still keeps her weight.*)
5	Và thực tế là cô ấy thật nhẹ nhàng!	entail	Cô ấy không nặng chút nào. (*She is not heavy at all.*)
6	Và thực tế là cô ấy thật nhẹ nhàng!	contra	Cô ấy rất hạnh phúc. (*She is very happy.*)
7	Vậy nên, tôi không có bất cứ câu chuyện cụ thể nào. (*Thus, I don't have any particular story.*)	entail	Tôi không có một cửa hàng cụ thể. (*I don't have any particular store.*)
8	Vậy nên, tôi không có bất cứ câu chuyện cụ thể nào	contra	Tôi có 1 cửa hàng riêng. (*I have my own store.*)
9	Vậy nên, tôi không có bất cứ câu chuyện cụ thể nào	neutral	Có rất nhiều cửa hàng. (*There are many stores.*)

The numbers in bold are the best score for each column *time to enter it all in.* respectively do not contradict or entail the premise sentence as being annotated.

The word "*nhẹ nhàng*" in the next three premise sentences (4, 5, 6) is incorrectly translated from its original English word *light*. The original sentence reads *And the fact is she was light!*, in which the word *light* should be translated to *nhẹ* (*not heavy at all*) rather than *nhẹ nhàng*, which means *sweet*, *graceful* or *soft*. An incorrect translation of a single word alone is enough to make all three hypothesis sentences wrongly annotated.

In a similar way, the English word *store* in samples 7, 8 and 9 is wrongly translated to *câu chuyện* which means *story*. This results in inconsistency of the corresponding annotated inference labels.

This analysis reveals that if the XNLI datasets were annotated more correctly, the meaning of sentences would be better preserved and integrating semantic information into the models may achieve a better score.

6 Conclusion

In this work, we have presented a method that incorporates explicit lexical and concept-level semantics to improve language inference. The semantic information is provided by the multilingual ConceptNet knowledge base. A thorough experimental study is conducted on four neural network architectures with state-of-the-art encoder models, including convolutional network, recurrent network and bidirectional transformer network. On average of the three strong models, our semantic-aware approach improves natural language inference in different languages.

There are several lines of research for future exploration. Firstly, we will seek a better way to exploit semantics information by combining lexical semantics with dependency semantics which comes from a dependency parser. We think that explicit semantic dependency between concepts of a sentence will be more fruitful than treating concepts as independent as in the current work. In a recent study, we have demonstrated the usefulness of syntactic structures in improving lexical embeddings (Dang & Le-Hong, 2021). Secondly, we plan to investigate an abstract meaning representation of whole sentences and seek a way to incorporate this hierarchical semantic information into the models. Thirdly, we would like to study the effect of semantic specialization for BERT (Lauscher et al., 2020) in the language inference task. In our current method, synonymy and antonymy information is indiscriminately combined in the bag of concepts, and this conflation can negatively impact model performance. We project that a finer-grained treatment of synonymy and antonymy would help improve further our current methods. Finally, we will investigate the accuracy of our proposed framework on other languages.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Research involving human and animal rights This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed consent Informed consent was not required as no human or animals were involved.

References

Bauer, L., Wang, Y., & Bansal, M. (2018). Commonsense for generative multi-hop question answering tasks. In *Proceedings of EMNLP, Brussels, Belgium* (pp. 4220–4230).

Bos, J., & Markert, K. (2005). Recognising textual entailment with logical inference. In *Proceedings of EMNLP, ACL, Brussels, Belgium* (pp. 628–635).

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP, ACL* (pp. 632–642).

Bowman, S. R., Potts, C., & Manning, C. D. (2015). Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China* (pp. 12–21).

Bui, T. V., Tran, T. O., & Le-Hong, P. (2020). Improving sequence tagging for Vietnamese text using transformer-based neural models. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation, Association for Computational Linguistics, Hanoi, Vietnam* (pp. 13–20).

Cambria, E., Fu, J., Bisio, F., & Poria, S. (2015). AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In *Proceedings of AAAI* (pp. 508–514).

Cambria, E., Li, Y., Xing, F., Poria, S., & Kwok, K. (2020). SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In *CIKM* (pp. 105–114).

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., E. R. H. Jr., & Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of AAAI* (pp. 10–18).

Chang, T.-Y., Liu, Y., Gopalakrishnan, K., Hedayatnia, B., Zhou, P., & Hakkani-Tur, D. (2020). Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Association for Computational Linguistics, Online* (pp. 74–79).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings NIPS 2014 Deep Learning and Representation Learning Workshop, Montreal, Canada* (pp. 10–19).

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR* (pp. 1–18).

Conneau, A., Rinott, R., Lample, G., Schwenk, H., Stoyanov, V., Williams, A., & Bowman, S. R. (2018). XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP, ACL, Brussels, Belgium* (pp. 2475–2485).

Dang, H.-V., & Le-Hong, P. (2021). A combined syntactic-semantic embedding model based on lexicalized tree-adjoining grammar. *Computer Speech and Language, 68*(2021), 101202.

de Marneffe, M.-C., Rafferty, A. N., & Manning, C. D. (2008). Finding contradictions in text. In *Proceedings of ACL, Columbus, Ohio, USA* (pp. 1039–1047).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL, Minnesota, USA* (pp. 1–16).

Fadel, A., Al-Ayyoub, M., & Cambria, E. (2020). JUSTers at SemEval-2020 task 4: Evaluating transformer models against commonsense validation and explanation. In *Proceedings of the 14th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Barcelona (online)* (pp. 535–542).

Fyodorov, Y., Winter, Y., & Francez, N. (2000). A natural logic inference system. In *Proceedings of the 2nd Workshop on Inference in Computational Semantics* (pp. 1–17).

Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague* (pp. 1–9).

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of EMNLP, ACL, Doha, Quatar* (pp. 1746–1751).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Y. Bengio, Y. LeCun (Eds.), Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA* (pp. 1–15).

Kyunghyun, C., van Merrienboer Bart, Caglar, G., Dzmitry, B., Fethi, B., Holger, S., & Yoshua, B. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv:1406.1078.

Lauscher, A., Majewska, O., Ribeiro, L. F. R., Gurevych, I., Rozanov, N., & Glavaš, G. (2020) Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Association for Computational Linguistics, Online* (pp. 43–49.

Lauscher, A., Vulić, I., Ponti, E. M., Korhonen, A., & Glavaš, G. (2020). Specializing unsupervised pretraining models for word-level semantic similarity. In *Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online)* (pp. 1371–1383).

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2015). Dbpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web, 6*(2), 167–195.

Le-Hong, P., Nguyen, T. M. H., Roussanaly, A., & Ho, T. V. (2008). A hybrid approach to word segmentation of Vietnamese texts. In *Language and Automata Theory and Applications, Vol. 5196 of Lecture Notes in Computer Science, Springer Berlin Heidelberg* (pp. 240–249).

Le-Hong, P., Roussanaly, A., Nguyen, T. M. H., & Rossignol, M. (2010). An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts. In *Actes de Traitement Automatique des Langues, Montreal, Canada* (pp. 50–61).

Li, Y., Pan, Q., Yang, T., Wang, S., Tang, J., & Cambria, E. (2020). Learning word representations for sentiment analysis. *Cognitive Computation, 9*, 843–851.

Liu, X., He, P., Chen, W., & Gao, J. (2019). Multi-task deep neural networks for natural language understanding. In *Proceedings of ACL, Florence, Italy* (pp. 4487–4496).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach.

Ma, Y., Peng, H., & Cambria, E. (2018). Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of AAAI* (pp. 5876–5883).

MacCartney, B., & Manning, C. D. (2009). An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics, ACL, Tilburg, The Netherlands* (pp. 140–156).

Mihaylov, T., & Frank, A. (2018). Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of ACL, Melbourne, Australia* (pp. 821–832).

Nguyen, D. Q., & Nguyen, A. T. (2020). PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 1037–1042).

Nguyen, M. -T., Ha, Q. -T., Nguyen, T. -D., Nguyen, T. -T., & Nguyen, L. -M. (2015). Recognizing textual entailment in vietnamese text: An experimental study. In *Proceedings of the Seventh International Conference on Knowledge and Systems Engineering (KSE), IEEE, Ho Chi Minh City, Vietnam* (pp. 108–113).

Peng, H., Cambria, E., & Hussain, A. (2017). A review of sentiment analysis research in Chinese language. *Cognitive Computation, 9*, 423–435.

Peters, M. E., Neumann, M., IV, R. L. L., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019) Knowledge enhanced contextual word representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Hongkong, China* (pp. 43–54).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL, Louisiana, USA* (pp. 1–15).

Pham, M. Q. N., Nguyen, M. L., & Shimazu, A. (2012). Using machine translation for recognizing textual entailment in Vietnamese language. In *Proceedings of IEEE International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, IEEE, Ho Chi Minh City, Vietnam* (pp. 1–6).

Poria, S., Cambria, E., Hussain, A., & Huang, G.-B. (2015). Towards an intelligent framework for multimodal affective data analysis. *Neural Networks, 63*, 104–116.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. In *Preprint* (pp. 1–12).

Satapathy, R., Cambria, E., Nanetti, A., & Hussain, A. (2020). A review of shorthand systems: From brachygraphy to microtext and beyond. *Cognitive Computation, 12*(4), 778–792.

Speer, R., Chin, J., & Havasi, C. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI 31, 2017* (pp. 4444–4451).

Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R. R., & Cohen, W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of EMNLP, Brussels, Belgium* (pp. 4231–4242).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30, pp. 5998–6008). Curran Associates Inc.

Wang, C., & Jiang, H. (2019). Explicit utilization of general knowledge in machine reading comprehension. In *Proceedings of ACL, Florence, Italy* (pp. 2263–82272).

Wang, C., Liang, S., Jin, Y., Wang, Y., Zhu, X., & Zhang, Y. (2020). SemEval-2020 task4: Commonsense validation and explanation. In *Proceedings of the 14th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Barcelona (online)* (pp. 307–321).

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium* (pp. 353—355).

Williams, A., Nangia, N., & Bowman, S. R. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT, ACL, New Orleans, Louisana, USA* (pp. 1112–1122).

Yang, B., & Mitchell, T. (2017). Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada* (pp. 1436–1446).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS* (pp. 5754–5764).

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., & Zhou, X. (2020). Semantics-aware BERT for language understanding. In *Proceedings of AAAI* (pp. 9628–9635).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.