# Learning to Differentiate Between Main-articles and Sub-articles in Wikipedia

Muhao Chen[1], Changping Meng[2], Gang Huang[3] and Carlo Zaniolo[1]
[1]*Department of Computer Science, University of California, Los Angeles*
[2]*Purdue University, West Lafayette*
[3]*Google Inc, Mountain View*
{muhaochen,zaniolo}@cs.ucla.edu;
ganghuang@google.com; meng40@purdue.edu

***Abstract*—Current Wikipedia editing approaches typically summarize a named entity by one *main-article* supplemented by multiple *sub-articles* describing various aspects and subtopics of the entity. Such separation of articles aims at improving the curation of content-rich Wikipedia entities. However, a wide range of Wikipedia-based technologies critically rely on the *article-as-concept* assumption, which requires a one-to-one mapping between entities (or concepts) and the articles that describe these entities. Thus, the current editing approaches sow confusion and ambiguity to knowledge representation, and cause problems to a wide-range of downstream technologies. In this paper, we present an approach that resolves these problems by differentiating the main-article from the sub-articles that are not at the core of entity representations. We propose a hybrid neural article model that learns on two facets of a Wikipedia article: (i) Two neural document encoders capture the latent semantic features from the article title and text contents. (ii) A set of explicit features measure and characterize the symbolic and structural aspects of each article. In this study, we use crowdsourcing to create a large annotated dataset for feature extraction, and for evaluating a variety of encoding techniques and learning structures. The optimized model so derived identifies main articles with near-perfect precision and recall, and outperforms various baselines on the contributed dataset.**

## I. Introduction

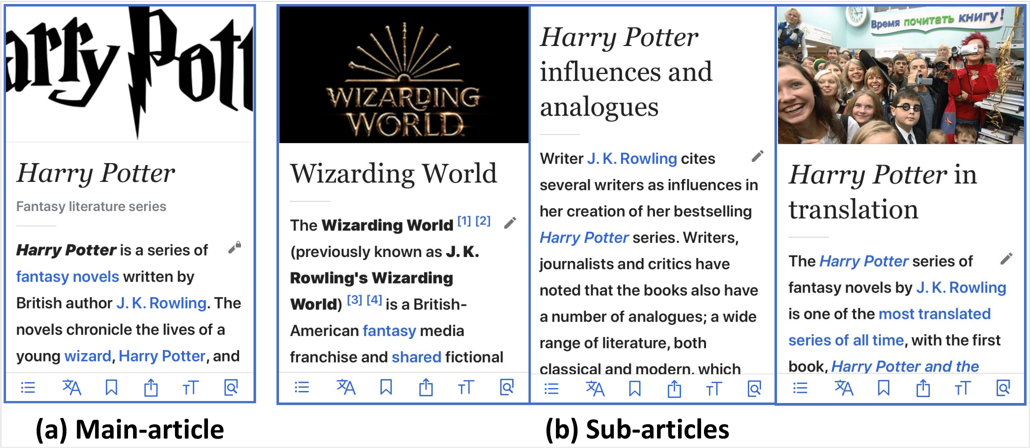
Wikipedia has been one of the most important sources of knowledge on the Web. This vast storage of more than 45 million encyclopedia articles for real-world entities has stimulated important computational research on knowledge base construction [20], commonsense reasoning [33], and natural language understanding [26]. It has also triggered countless content management and retrieval technologies for semantic search [24], question answering [4], and data linage [42].

Most aforementioned technologies essentially rely on a key assumption called *article-as-concept* [23, 21, 7], which regulates a one-to-one mapping between entities (or concepts) and the associated Wikipedia articles. However, this assumption is being nullified by the recent trends of editing in Wikipedia, which encourage editors to divide entities with rich contents into multiple articles that will later be managed independently [1, 23]. Such content splitting process typically places the contents into one *main-article* for the general summarization of a named entity, and multiple *sub-articles* for the description of specific aspects and subtopics of the entity. For example, as shown in Fig. 1, the entity *Harry Potter* is provided with an overall introduction in its main-article, while some content details of its sub-topics can also be found in its sub-articles *Wizarding World*, *Harry Potter Influences* and *Analogues, and Harry Potter in Translation*. Because the split-off of Wikipedia articles has shown effectiveness in improving the curation of rich contents of entities, corresponding editing is now frequent in Wikipedia. Recent research has indicated that over 70% of the most popular 1,000 Wikipedia entities have split-offs [23].

Although separating main and sub-articles benefits Wikipedia users with better content curation, violation of the article-as-concept assumption causes many deleterious effects on Wikipedia-based technologies from the following three perspectives. (i) *Ambiguous entity representations.* Separating contents of one entity into articles with different titles is problematic to Wikipedia-based knowledge base construction [20, 37], which usually assume the article title as an entity name and extract relational facts and entity descriptions from the article contents. Split-off of articles no doubt sow confusion in corresponding processes. Meanwhile, it also hinders downstream technologies such as named entity recognition [28], document relatedness analysis [13] and question answering [4], which are all based on the representations of entities that are free from ambiguity. (ii) *Diffused entity descriptions.* Diffusing entity descriptions across multiple articles seriously affects semantic search of entities [24], which require the general summarizations of those entities to be identified from the descriptions on sub-topics. Linage of entities [42] is also hard to track, since the contents of an entity now span across multiple pages. (iii) *Cross-lingual inconsistency.* Multilingual tasks such as knowledge alignment [6, 8, 30, 34], content synchronization [3] and bilingual Wikification [36] are also challenged, since these tasks typically assume one-to-one match of articles across language versions of Wikipedia.

To support these crucial technologies, it is highly demanded to address the *main-article identification problem*, which seeks to automatically differentiate between the main and sub-articles in Wikipedia. Through the resolve of this problem, the article-as-concept assumption can be restored through the 1-to-1 mapping between main-articles and the entities they

**978-1-7281-0858-2/19/$31.00 ©2019 IEEE**



*Harry Potter*

Fantasy literature series

***Harry Potter*** is a series of fantasy novels written by British author J. K. Rowling. The novels chronicle the lives of a young wizard, Harry Potter, and

Wizarding World

The **Wizarding World** [1] [2] (previously known as **J. K. Rowling's Wizarding World**) [3] [4] is a British-American fantasy media franchise and shared fictional

*Harry Potter* influences and analogues

Writer J. K. Rowling cites several writers as influences in her creation of her bestselling *Harry Potter* series. Writers, journalists and critics have noted that the books also have a number of analogues; a wide range of literature, both classical and modern, which

*Harry Potter* in translation

The *Harry Potter* series of fantasy novels by J. K. Rowling is one of the most translated series of all time, with the first book, *Harry Potter and the*

**(a) Main-article**

**(b) Sub-articles**

Fig. 1: Main and sub-articles of entity *Harry Potter*.

summarize. However, it is a non-trivial mission for a model to precisely capture the distinction of these two styles of articles for several reasons: (i) This objective requires a model to identify the summary style of main-article contents, which is invariant to the topics and meanings of articles. (ii) The model should also differentiate those from sub-articles that are semantically similar to main-articles with shared content details. (iii) In addition to text contents, the characteristics of both articles can also be reflected from different components of Wikipedia articles, including titles, links and article structures. Besides, a large collection of labelled main and sub-articles are also needed for the extraction of features to capture the style differences of the two article types.

We propose a model that addresses the main-article identification problem. The proposed model employs a hybrid learning architecture that characterizes two facets of a Wikipedia article: (i) Two neural document encoders respectively capture the latent semantic features from the article titles and text contents, for which we extensively explore with a wide variety of encoding techniques and learning structures. (ii) A comprehensive set of explicit features are also used to measure the symbolic and structural aspects of the articles. A large crowdsourced dataset is created to support the evaluation and feature extraction for the task. Held-out estimation on the dataset distinguishes the proper document encoding techniques, and proves the effectiveness of the best model variants for achieving near-perfect precision and recall for identifying main-articles, and significantly outperform various baselines. Ablation study also offers insights in feature selection for characterizing Wikipedia main and sub-articles.

## II. Related Work

The violation of the article-as-concept assumption in Wikipedia have been problematized very recently [23]. A few recent works have focused on the *sub-article matching problem*, which aims at identifying the association between main and sub-articles based on candidate article pairs. Corresponding solutions are relevant to the learning of article pair models for discourse relations [31, 41]. To capture the relation of article pairs, Lin et al. [23] train several statistical learning algorithms on a set of handcrafted features. These features focus on measuring the symbolic similarity, relative centrality and cross-lingual co-occurrence of these article pairs in Wikipedia. Chen et al. [7] propose a hierarchical learning architecture that incorporates neural article pair models with the features in [23]. This approach has shown the state-of-the-art performance on detecting the match of main and sub-articles on large collections of candidate article pairs. However, the sub-article matching problem relies on the premise that main-articles and sub-articles can be differentiated between, so as to generate valid candidates for the corresponding models. This is exactly the focus of the main-article identification problem which we seek to address.

Research on neural text classification represents one fundamental problem of natural language processing. Tasks based on this problem normally span in two categories, i.e. sentiment classification and topic classification [45, 43]. Downstream classifiers for corresponding tasks rely on successfully extracting the semantic features from text, and many recent efforts adopt different forms of deep neural document encoders for this purpose. These encoders typically employ CNN [18] or variants of RNN [38, 44] to aggregate the lexical semantic features and sequence information of sentences. On top of these, some works introduce attention mechanisms, which seek to highlight the lexical semantics that are important to the overall meanings or sentiments of sentences and paragraphs [38, 5]. Hierarchical [35, 45] and hybrid [19, 15] learning structures of neural sentence encoders are proposed to aggregate the semantic features of multiple sentences, which achieves state-of-the-art performance for sentiment and topic classification of long paragraphs. The main-article identification problem entails the semantic representations of text contents and titles of Wikipedia articles. However, while we have experimented with a wide variety of neural document classification techniques, we show that successfully identifying main-articles is rather challenging with the latent semantic features alone. Unlike sentiments and topics that are solely implied by the semantic meanings of text, the summary style of main-articles to be comprehended by the model is invariant to topics and meanings of the article contents. And this challenging problem also depends on comprehensively characterizing other aspects of Wikipedia articles, such as their hyperlink structures, section structures and symbolic measures.

## III. Modeling

We hereby introduce our model for Wikipedia main-article identification. We begin with denotations and the problem definition.

### A. Preliminaries

**Denotations.** Denotations of Wikipedia articles are coherent to those in [7]. We use $W$ to denote the set of Wikipedia articles. An article $A_i \in W$ is modeled as a triple $A_i = (t_i, c_i, o_i)$, such that $t_i$ is the title, $c_i$ the text contents, and $o_i$ the miscellaneous structural information such as templates, sections and links. $t_i = w_{t1}, w_{t2}, ..., w_{tl}$ and $c_i = w_{c1}, w_{c2}, ..., w_{cm}$ thereof are both sequences of lexicons. $c_i$ is also partitioned into multiple sentences, i.e. $c_i = s_1 \oplus s_2 \oplus ... \oplus s_n$. For simplicity, we use $s_j \subset c_i$ to denote that $s_j$ is a sentence of $c_i$, which is a consecutive subsequence of $c_i$. In practice, we use the first paragraph of $A_i$ to represent $c_i$, since it is the summary of the article contents. For each word $w$, we use bold-faced $\mathbf{w}$ to denote its embedding representation. We use $F(A_i)$ to denote a series of explicit features that provide some symbolic and structural measures for titles, text contents and link structures of $A_i$, which we are going to specify in Section III-C. We assume that all articles to be differentiated by the model is written in the same language, as a separated solution can be developed for each language version of Wikipedia. In this paper, we only consider English articles *w.l.o.g*.

**Problem definition.** *Main-article identification* is defined as a binary classification problem on the Wikipedia article set $W$. Given an article $A_i \in W$, a model should decide whether $A_i$ is a main-article or not. Note that this set of articles excludes those that belong to a meta-article category such as *Lists* [1] and *Disambiguation* [2], which are considered to be neither of the two article types of our interest. $A_i$ is a main-article if $t_i$ is a stand-alone entity name, and $c_i$ generally summarizes most aspects of the entity name $t_i$. Otherwise, $A_i$ is a sub-article. To address the main-article identification problem, our model learns on a combination of two aspects of each Wikipedia article. Neural document encoders extract the latent semantic features from text, while a series of explicit features are incorporated to characterize the symbolic or structural aspects. In the following, we introduce each component of our model in detail.

### B. Neural Document Encoders

We investigate with multiple variations of neural document encoders. The basic plain encoders capture the latent semantic feature from a sequence of lexicons, which can be used to represent the title and the text contents of a given article. The hierarchical and hybrid hierarchical encoders consider the paragraphical structures of text contents, and provide a multi-granular composition of semantics.

*1) Plain Encoders:* A plain document encoder $E_P(X)$ encodes a sequence of lexicons $X$ into an embedding vector of a sentence or the whole text contents of an article. We explore with three widely used encoding techniques that form the plain encoder, i.e., the convolutional encoders (CNN), the gated recurrent unit encoders (GRU), and self-attentive GRU encoder (ATT).

**Convolutional encoders.** A CNN employs the 1-dimensional convolution layer to encode an input sequence [7]. Given the input sequence $X = \{w_1, w_2, ..., w_l\}$, a convolution layer applies a kernel $\mathbf{M}_c \in \mathbb{R}^{h \times k}$ to generate a latent representation $\mathbf{h}_i^{(1)}$ from a window of the input vector sequence $\mathbf{w}_{i:i+h-1}$ by

$$\mathbf{h}_i^{(1)} = \tanh(\mathbf{M}_c \mathbf{w}_{i:i+h-1} + \mathbf{b}_c)$$

for which $h$ is the kernel size and $\mathbf{b}_c$ is a bias vector. The convolution layer applies the kernel to all consecutive windows to produce a sequence of latent vectors $\mathbf{H}^{(1)} = \mathbf{h}_1^{(1)} \oplus \mathbf{h}_2^{(1)} \oplus ... \oplus \mathbf{h}_{l-h+1}^{(1)}$, where each latent vector leverages the significant local semantic features from each $h$-gram of the input sequence. Like many other works [18, 16, 32], we apply $n$-max-pooling to extract robust features from each $n$-stride of the convolution outputs by $\mathbf{h}_i^{(2)} = \max(\mathbf{h}_{i:n+i-1}^{(1)})$, while the last layer uses a global average pooling [22] to obtain a vector representation of the input sequence.

**Gated Recurrent Units.** The GRU encoder is an alternative to LSTM [9], which consecutively characterizes sequence information without using separated memory cells [7]. Each unit consists of two types of gates to track the state of the sequence, i.e. the reset gate $\mathbf{r}_t$ and the update gate $\mathbf{z}_t$. Given the vector representation $\mathbf{w}_t$ of an incoming item $w_t$, GRU updates the hidden state $\mathbf{h}_t^{(3)}$ of the sequence as a linear combination of the previous state $\mathbf{h}_{t-1}^{(3)}$ and the candidate state $\tilde{\mathbf{h}}_t^{(3)}$ of new item $w_t$, which is calculated as below.

$$\mathbf{h}_t^{(3)} = \mathbf{z}_t \odot \tilde{\mathbf{h}}_t^{(3)} + (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1}^{(3)}$$

The update gate $\mathbf{z}_t$ balances between the information of the previous sequence and the new item, where $\mathbf{M}_z$ and $\mathbf{N}_z$ are two weight matrices, $\mathbf{b}_z$ is a bias vector, and $\sigma$ is the sigmoid function. The candidate state $\tilde{\mathbf{h}}_t^{(3)}$ is calculated similarly to those in a traditional recurrent unit. And the reset gate $\mathbf{r}_t$ controls how much information of the past sequence should contribute to $\tilde{\mathbf{h}}_t^{(3)}$.

$$\begin{aligned}
\mathbf{z}_t &= \sigma\left(\mathbf{M}_z \mathbf{w}_t + \mathbf{N}_z \mathbf{h}_{t-1}^{(3)} + \mathbf{b}_z\right) \\
\tilde{\mathbf{h}}_t^{(3)} &= \tanh\left(\mathbf{M}_s \mathbf{w}_t + \mathbf{r}_t \odot (\mathbf{N}_s \mathbf{h}_{t-1}^{(3)}) + \mathbf{b}_s\right) \\
\mathbf{r}_t &= \sigma\left(\mathbf{M}_r \mathbf{w}_t + \mathbf{N}_r \mathbf{h}_{t-1}^{(3)} + \mathbf{b}_r\right)
\end{aligned}$$

The above defines a GRU layer which outputs a sequence of hidden state vectors given the input sequence $X$. While a

[1] https://en.wikipedia.org/wiki/Category:Lists

[2] https://en.wikipedia.org/wiki/Wikipedia:Disambiguation

GRU encoder can consist of a stack of multiple GRU layers, without an attention mechanism, the last hidden state $\mathbf{h}_X^{(3)}$ of the last layer is extracted to represent the overall meaning of the encoded sequence. Note that in comparison to GRU, the traditional LSTM usually performs comparably, but is more complex and require more computational resources for training [10].

**Self-attention.** The self-attention mechanism [11] seeks to capture the overall meaning of the input sequence unevenly from each encoded item. One layer of self-attention is calculated as below.

$$\begin{aligned}
\mathbf{u}_t &= \tanh\left(\mathbf{M}_a \mathbf{h}_t^{(3)} + \mathbf{b}_a\right) \\
a_t &= \frac{\exp\left(\mathbf{u}_t^\top \mathbf{u}_X\right)}{\sum_{w_m \in X} \exp\left(\mathbf{u}_m^\top \mathbf{u}_X\right)} \\
\mathbf{h}_t^{(4)} &= |X| a_t \mathbf{u}_t
\end{aligned}$$

$\mathbf{u}_t$ thereof is the intermediary latent representation of the GRU output $\mathbf{h}_t^{(3)}$, and $\mathbf{u}_X = \tanh(\mathbf{M}_a \mathbf{h}_X^{(3)} + \mathbf{b}_a)$ is the intermediary latent representation of the last GRU output $\mathbf{h}_X^{(3)}$ that can be seen as a high-level representation of the entire input sequence. By measuring the similarity of each $\mathbf{u}_t$ with $\mathbf{u}_X$, the normalized attention weight $a_t$ for $\mathbf{h}_t^{(3)}$ is produced through a softmax function, which highlights an input that contributes more significantly to the overall meaning. Note that a scalar $|X|$ (the length of the input sequence) is multiplied along with $a_t$ to $\mathbf{u}_t$ to obtain the weighted representation $\mathbf{h}_t^{(4)}$, so as to keep $\mathbf{h}_t^{(4)}$ from losing the original scale of $\mathbf{h}_t^{(3)}$. A latent representation of the sequence is calculated as the average of the last attention layer $E_P^{(2)}(X) = \frac{1}{|X|} \sum_{t=1}^{|X|} a_t \mathbf{h}_t^{(4)}$.

*2) Hierarchical and Hybrid Encoders:* A *hierarchical encoder* provides a multi-granular encoding process for an article using two levels of plain encoders. A sentence-level encoder $E_P^{(s)}$ first aggregate the lexical semantics for each sentence $s_j \subset c_i$. Note that the parameters of $E_P^{(s)}$ are shared among the encoding process of all sentences in an article. On top of that, an article-level plain encoder $E_P^{(a)}$ is used to combine the encodings of all sentences into the latent semantic representation of the whole text contents. Hence, the embedding of the given $c_i$ is obtained as follows:

$$E_H(c_i) = E_P^{(a)}\left(\bigoplus_{s_j \subset c_i} E_P^{(s)}(s_j)\right)$$

Different types of $E_P^{(a)}$ correspond to different purposes for composing the sentential semantics. CNN thereof, seeks to preserve the local interactions of consecutive sentences [17], while GRU and attentive GRU focus on capturing the sequence of sentential semantics, and highlighting the important sentences that have overall more contributions to the article topics [35, 39]. Adopting the same one of the three encoding techniques for $E_P^{(s)}$ and $E_P^{(a)}$ lead to three variants of regular hierarchical encoders. Meanwhile, six types of *hybrid hierarchical encoders* (or simply, *hybrid encoders*) employ differently the techniques for $E_P^{(s)}$ and $E_P^{(a)}$. Details of the model variants led by different forms of hierarchical encoders are described in Section IV-B.

### C. Explicit Features

In addition to the latent semantic features captured by neural document encoders, we define a set of explicit features $F(A_i) = \{f_{self}, n_{tps}, n_{sen}, n_{sec}, r_{tto}, r_{cto}, r_{sto}, d_{in}, d_{out}\}$ for an article $A_i$:

- $f_{self}$: self-mentioning ratio, which is defined as the term-frequency of title $t_i$ in text content $c_i$.
- $n_{tps}$: the average number of tokens per sentence in $c_i$, which provides a verbosity measure of the article.
- $n_{sen}$: the number of sentences in $c_i$.
- $n_{sec}$: the number of sections $A_i$ is divided into.
- $r_{tto}$: the maximum token overlap ratio of $t_i$ with titles of other articles.
- $r_{cto}$: the maximum token overlap ratio of $t_i$ with section titles of other articles.
- $r_{sto}$: the maximum token overlap ratio of section titles of $A_i$ with titles of other articles.
- $d_{in}$: in-degree centrality of $A_i$ based on inline hyperlinks.
- $d_{out}$: out-degree centrality of $A_i$ based on inline hyperlinks.

We normalize these features via feature scaling [3], except for the self-mentioning ratio $f_{self}$ that has already been normalized by definition. These features comprehensively measure the symbolic and structural aspects of a Wikipedia article. As shown in our experiments, these features alone are able to provide effective characterization of the main and sub-articles categories, and are key to enhance the characterization by the document encoder.

### D. Learning Objective

We use two neural document encoders $E_t$ and $E_c$ to encode the title and text contents of $A_i \in W$ respectively. Both neural document encoders first employ a pre-trained lexicon embedding layer that captures the lexical semantics in the embedding space. $E_t$ thereof is a plain document encoder, while $E_c$ can be a plain encoder or a hierarchical encoder. Note that in cases where a hierarchical $E_c$ is adopted, the sentence-level encoder $E_P^{(s)}$ utilizes the same encoding technique of $E_t$, while the article-level $E_P^{(a)}$ may utilize a different encoding technique to form a hybrid encoder based on different assumptions of aggregations for sentential semantics. Two sigmoid multi-layer perceptrons (MLP) are then applied to $E_c(c_i)$ and $E_t(t_i)$ to produce two confidence scores $z_{c_i}$ and $z_{t_i}$ to support $A_i$ to be a main-article. Then we concatenate the explicit features $F(A_i)$ along with the two confidence scores, i.e. $z_{c_i} \oplus z_{t_i} \oplus F(A_i)$. On top of that, another set of linear MLP is applied to obtain the two confidence scores $\hat{y}_{A_i}^+$ and $\hat{y}_{A_i}^-$ for the boolean labels of positive identification $l^+$ and negative identification $l^-$ respectively. Lastly, $\hat{y}_{A_i}^+$ and $\hat{y}_{A_i}^-$ are normalized by binary softmax functions $y_{A_i}^+ = \frac{\exp(\hat{y}_{A_i}^+)}{\exp(\hat{y}_{A_i}^+) + \exp(\hat{y}_{A_i}^-)}$

[3] https://en.wikipedia.org/wiki/Feature_scaling

and $y^-_{A_i} = \frac{\exp(\hat{y}^-_{A_i})}{\exp(\hat{y}^+_{A_i})+\exp(\hat{y}^-_{A_i})}$. The learning objective is to minimize the following binary cross-entropy loss.

$$L = -\frac{1}{|W|}\sum_{A_i \in W}\left(l^+ \log y^+_{A_i} + l^- \log y^-_{A_i}\right)$$

## IV. Experiments

In this section, we present the experimental evaluation of the proposed approaches. We first create a large dataset for the Wikipedia main-article identification problem via massive crowdsourcing. Using this dataset, we comprehensively evaluate several categories of model variants and baselines based. In addition, we present an ablation study to help the feature selection for this task.

### A. Dataset Preparation

We prepare the dataset through the internal crowdsourcing platform of *anonymous corporation*. To first produce some candidate articles for crowdsourced annotation, a set of candidate sub-articles whose titles concatenate two Wikipedia entity names directly or with a proposition are selected, e.g. *French Economy* and *Censorship in Finland*. Selection of such candidate articles is based on the hypothesis that corresponding formations of titles are more likely to be those of sub-articles. Then we sample from this set of articles for annotation in the crowdsourcing platform and instruct the annotators to decide whether each sampled candidate is a sub-article. If so, the annotators will be asked to provide the main-article for the recognized sub-article. The matching of main and sub-articles is qualified based on two criteria, i.e. $A_j$ is a sub-article of $A_i$ if (i) $A_j$ describes an aspect or a subtopic of $A_i$, and (ii) $c_j$ can be inserted as a section of $A_i$ without breaking the topic summarized by $t_i$. Each crowdsourced article has been reviewed by three annotators, and is populated into the dataset if total agreement is reached. This process has populated the dataset with around 22 thousand articles, where 5,012 are main-articles, and 17,349 are sub-articles.

### B. Main-article Identification

We evaluate the proposed model variants based on held-out estimation. These model variants are classified into three categories:

(i) Model variants combining explicit features with three types of plain document encoders, i.e. CNN+$F$, GRU+$F$ and ATT+$F$.

(ii) Model variants combining explicit features with three types of regular hierarchical encoders, i.e. HCNN+$F$, HGRU+$F$ and HATT+$F$, each of which employs the same encoding technique for sentence/title encoders and the article-level aggregation.

(iii) Those combining explicit features with six different hybrid encoders, each of which employs one type of encoding technique for sentences and titles, and another type for article-level aggregation of sentence encodings. Different encoding techniques are denoted separately for the hybrid encoders, such that we use the superscripts $(t, s)$ to mark the encoding technique used for sentences and titles, and $(a)$ to mark that for article-level aggregation. For example, CNN$^{(t,s)}$+GRU$^{(a)}+F$ denotes the model where CNN is used for titles and sentences, while GRU is used for the article-level aggregation of sentential semantics for the text contents, and explicit features are also incorporated.

We compare with neural document classifiers without explicit features that represent the line of related work for document classification [18, 38, 44, 35, 43, 15]. We have also experimented with the much simpler linear bag-of-words encoder [14], which is not taken into comparison for its being substantially outperformed by other encoding techniques. Besides, we also compare with statistical learning algorithms that solely rely on the explicit features.

**Model configurations.** We use AMSGrad [29] to optimize the learning objective loss, for which we set the learning rate $\alpha$ to 0.001, the exponential decay rates $\beta_1$ and $\beta_2$ to 0.9 and 0.999, and batch size to 64. As we have stated in Section III-A, we encode the first paragraph of each article to represent its text contents. When inputting text contents into plain document encoders, we remove stop words in the text contents, zero-pad short ones and truncate overlength ones to the sequence length of 150. Hierarchical encoders delimit the number of sentences to 9, and zero-pad/truncate each sentence to the sequence length of 20. Such length-normalized representations allow most (over 92%) of lexicons to be fed into the document encoders. We also zero-pad short titles to the sequence length of 14, which is the maximum length of the original titles. For the implementation of neural document encoders, we use two layers of the corresponding encoder type for titles, three for text contents in models that adopt only plain encoders, and two for sentence encoding and one for article-level aggregation in models with a hierarchical or hybrid encoder. The dimensionality of document embeddings is selected among {100, 150, 200, 300}, for which we fix 100 for titles and 200 for text contents. For CNN, we select kernel sizes and pool sizes from 2 to 4, with the kernel size of 3 and 2-max-pooling adopted. In addition, we use one hidden layer in MLPs where the hidden-layer size is the average of the input and output layer sizes following the convention [40, 2, 7].

For the lexicon embedding layer, we pre-train the Skip-Gram [25] on the English Wikipedia dump. We use context size of 20, minimum word frequency of 7 and negative sampling size of 5 to obtain 150-dimensional lexicon embeddings. After pre-training, we fix the lexicon embeddings to convert titles and text contents to sequences of vectors to be fed into the neural document encoder.

**Evaluation Protocols.** We adopt 10-fold cross-validation in the evaluation process following previous works of document classification [18, 38]. At each fold, all models are trained till convergence. Since the objective of this task is to identify the relatively rare main-article type, we aggregate *precision*, *recall* and *F1-scores* on the positive cases at each fold of testing.

**Results.** Results by all model variants and baselines are

TABLE I: Evaluation results for main-article identification. We report precision, recall and F1-scores (all in percentages) for five groups of models: (1) Three baseline neural document classifiers with plain encoders and nine with hierarchical or hybrid encoders; (2) Baselines of statistical classification algorithms based on explicit features, including Logistic Regression, Naive Bayes Classifier (NBC), Linear SVM, Adaboost (SAMME.R algorithm [46]), Decision Tree (DT) and Random Forest (RF); (3) The proposed Model variants that combine explicit features with plain encoders, i.e. CNN+$F$, GRU+$F$ and ATT+$F$; (4) Those incorporating explicit features in hierarchical encoders, i.e. HCNN+$F$, HGRU+$F$ and HATT+$F$; (5) Model variants incorporating explicit features in hybrid encoders, e.g. CNN$^{(t,s)}$+GRU$^{(a)}$ + $F$.

| Group (1) | | | | | | | | Group (2) | | | | Groups (3) & (4) | | | | Group (5) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Models | Prec | Recall | F1 | Models | Prec | Recall | F1 | Models | Prec | Recall | F1 | Models | Prec | Recall | F1 | Models | Prec | Recall | F1 |
| CNN | 56.09 | 56.43 | 56.26 | CNN$^{(t,s)}$+GRU$^{(a)}$ | 59.14 | 51.86 | 55.26 | Logistic | 82.48 | 91.63 | 86.82 | CNN+$F$ | 92.72 | 90.48 | 91.59 | CNN$^{(t,s)}$+GRU$^{(a)}$ + $F$ | 96.04 | 95.67 | 97.34 |
| GRU | 60.46 | 51.48 | 55.61 | CNN$^{(t,s)}$+ATT$^{(a)}$ | 60.73 | 49.73 | 54.68 | NBC | 49.93 | 41.42 | 45.28 | GRU+$F$ | 95.45 | 84.66 | 89.73 | CNN$^{(t,s)}$+ATT$^{(a)}$ + $F$ | 97.13 | 93.84 | 95.46 |
| ATT | 59.31 | 52.67 | 55.79 | GRU$^{(t,s)}$+CNN$^{(a)}$ | 59.66 | 58.18 | 58.91 | Adaboost | 89.79 | 85.72 | 87.71 | ATT+$F$ | 95.25 | 86.89 | 90.88 | GRU$^{(t,s)}$+CNN$^{(a)}$ + $F$ | 98.31 | **99.01** | **98.66** |
| HCNN | 56.80 | 55.59 | 56.19 | GRU$^{(t,s)}$+ATT$^{(a)}$ | 62.83 | 52.42 | 57.15 | Linear SVM | 86.76 | 85.91 | 86.34 | HCNN+$F$ | **98.72** | 90.47 | 94.42 | GRU$^{(t,s)}$+ATT$^{(a)}$ + $F$ | 97.78 | 92.53 | 95.08 |
| HGRU | 56.90 | 52.75 | 54.75 | ATT$^{(t,s)}$+CNN$^{(a)}$ | 61.28 | 58.80 | 60.01 | DT | 83.28 | 79.56 | 81.37 | HGRU+$F$ | 97.95 | 88.46 | 92.96 | ATT$^{(t,s)}$+CNN$^{(a)}$ + $F$ | 98.53 | **98.84** | **98.68** |
| HATT | 61.75 | 51.90 | 56.40 | ATT$^{(t,s)}$+GRU$^{(a)}$ | 59.84 | 53.89 | 56.71 | RF | 89.48 | 88.68 | 89.08 | HATT+$F$ | 98.17 | 88.61 | 93.15 | ATT$^{(t,s)}$+GRU$^{(a)}$ + $F$ | **98.76** | 94.65 | 96.66 |

TABLE II: Ablation on feature categories for ATT$^{(t,s)}$+CNN$^{(a)}$ + $F$.

| Features | Precision | Recall | F1 |
|---|---|---|---|
| All features | 98.53 | 98.84 | 98.68 |
| Remove titles | 97.73 | 85.85 | 91.41 |
| Remove text contents | 97.97 | 91.67 | 94.71 |
| Remove explicit | 61.28 | 58.80 | 60.01 |

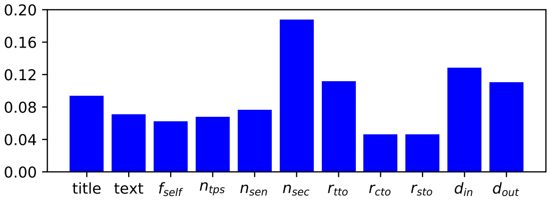


Fig. 2: Relative importance (RI) of features analyzed by Garson's algorithm.

reported in Table I. It is noteworthy that, neural document classifiers, which have been widely used in sentiment and topic-based document classification tasks [18, 38, 44, 35, 43, 15], fall short of effectively identifying the main-articles from candidate Wikipedia articles based on only semantic features. This shows that the summary style of main-articles is invariant to topics and meanings of the article contents. On the other hand, the explicit features alone are helpful to the task. The best statistical classification algorithm (Random Forest) is able to outperform the best neural document classifier that is solely based on semantic features (ATT$^{(t,s)}$+CNN$^{(a)}$) drastically by 29.07% in F1-score. This indicates that the explicit features that measure the symbolic and structural aspects of the Wikipedia articles are critical to distinguish the main-articles.

Meanwhile, combining both latent semantics and explicit features have significant enhancement on addressing this task, such that the proposed model variants that combine both categories of features generally obtain much higher F1 than the best baseline on Random Forest. For model variants with different encoding architectures, we notice that those with hierarchical encoders consistently outperform those with simpler plain document encoders, such that the best regular hierarchical model HCNN+$F$ improves the best plain-encoder-based CNN+$F$ by 2.83% of F1. Models that employ hybrid encoders offer even better performance, among which the best performance is achieved in the cases where GRU or ATT is employed for title and sentence encoding, and CNN is used for article-level aggregation of sentence representations. Corresponding model variants GRU$^{(t,s)}$+CNN$^{(a)}$ + $F$ and ATT$^{(t,s)}$+CNN$^{(a)}$ + $F$ achieve near-perfect F1 scores and outperform HCNN+$F$ and Random Forest by around 4.2% and 9.6% of F1 respectively, as well as with both notably higher precision and recall. This indicates that the best article representation strategy for this task requires sequence encoders to capture the sequence information of titles and short sentences, while the local interactions of adjacent sentences preserved by the article-level CNN are more important than the sequence of these sentences. This also explains why the GRU$^{(t,s)}$+CNN$^{(a)}$ and ATT$^{(t,s)}$+CNN$^{(a)}$ without explicit features outperform the other neural document classifier baselines in group (1) of Table I. The effect of self-attention mechanism for title and sentence-level encoders thereof, is relatively marginal. To summarize, the proposed best model variants have achieved very promising performance on addressing the main-article identification problem.

### *C. Ablation Study of Features*

Next, we perform the ablation study on different categories of features and each individual feature, so as to analyze their significance to the task. Table II shows the ablation of feature categories for the best model variant ATT$^{(t,s)}$+CNN$^{(a)}$ + $F$. We have already shown that removing the explicit features would significantly impair the performance of the model. As for the two categories of latent semantic features based on titles and text contents, we find that removing either of them would noticeably impair the model performance in terms of recall. The removal of title embeddings thereof cause more significant drop of performance than that of text content embeddings.

To understand the relative importance (RI) of each specific feature, we process Garson's weight analysis algorithm [27, 12] on the last linear MLP of ATT$^{(t,s)}$+CNN$^{(a)}$ + $F$. Fig. 2 shows the RI of the individual features, which is aggregated from all folds of cross-validation. The number of sections $n_{sec}$ thereof appear to be the most important feature for characterizing the main and sub-article types. This is coherent to the fact that main-articles are typically divided into more sections for different aspects of the described entities, some of which are further extended to sub-articles. Besides, the title-based semantic feature and $r_{tto}$ as well as degree-centrality-based $d_{in}$ and $d_{out}$ also show high RI. This indicates that titles and link structures have higher significance than other aspects like text contents and section titles for characterizing the Wikipedia article in our task. These features also reflect the fact that, semantically it is easy to determine the article type based on titles without the text contents of articles, as it is close to human practice. Also, the main-articles are essentially characterized by its significance in the hyperlink structure.

## V. Conclusion

In this paper, we have proposed a deep neural model for identifying main and sub-article in Wikipedia. The proposed model employs a wide varieties of document encoders for titles and text contents to capture the latent semantic features of Wikipedia articles, which is also incorporated with a set of explicit features to comprehensively measure the symbolic and structural aspects of articles. We have prepared a large dataset for the main-article identification problem via massive crowdsourcing. Held-out estimation shows that the proposed model variants significantly outperform existing neural document classification models based solely on semantic features, as well as statistical classification algorithms based on explicit features. The best model variants which employ hybrid document encoders and explicit features are able to achieve near-perfect performance. Meanwhile, ablation study of features have analyzed the relative importance of different feature categories and different individual features for our task.

## VI. Acknowledgement

This work is done during MC's and CM's summer research internships at Google, Mountain View. We thank the anonymous reviewers for their insightful comments.

## References

[1] P. Bao, B. Hecht, S. Carton, M. Quaderi, M. Horn, and D. Gergle, "Omnipedia: bridging the wikipedia language gap," in *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 2012.

[2] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, 2009.

[3] A. Bronner, M. Negri, Y. Mehdad, A. Fahrni, and C. Monz, "Cosyne: Synchronizing multilingual wiki content," in *Proceedings of the International Symposium on Open Collaboration*, 2012.

[4] D. Chen, A. Fisch *et al.*, "Reading Wikipedia to answer open-domain questions," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2017.

[5] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2016.

[6] M. Chen, Y. Tian *et al.*, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proceedings of the International Conference on Artificial Intelligence*, 2017.

[7] M. Chen, C. Meng, G. Huang, Z. Xue, and C. Zaniolo, "Neural article pair modeling for wikipedia sub-article matching," in *Proceedings of the European Conference on Machine Learning and Principles of Knowledge Discovery in Databases*. Springer, 2018.

[8] M. Chen, Y. Tian, K.-W. Chang, S. Skiena, and C. Zaniolo, "Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment," in *Proceedings of the International Conference on Artificial Intelligence*, 2018.

[9] K. Cho, B. van Merrienboer, C. Gulcehre *et al.*, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2014.

[10] J. Chung, C. Gulcehre *et al.*, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv*, 2014.

[11] A. Conneau, D. Kiela, H. Schwenk *et al.*, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2017.

[12] R. Féraud and F. Clérot, "A methodology to explain neural network classification," *Neural Networks*, vol. 15, no. 2, 2002.

[13] Z. Fu, F. Huang, X. Sun, A. Vasilakos, and C.-N. Yang, "Enabling semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Transactions on Service Computing*, 2016.

[14] F. Hill, K. Cho, A. Korhonen *et al.*, "Learning to understand phrases by embedding the dictionary," *Transactions of the Association for Computational Linguistics*, 2016.

[15] S. T. Hsu, C. Moon, P. Jones, and N. Samatova, "A hybrid cnn-rnn alignment model for phrase-aware sentence classification," in *Proceedings of the 2017 European Conference on Machine Learning*, 2017.

[16] B. Hu, Z. Lu *et al.*, "Convolutional neural network architectures for matching natural language sentences," in *Advances in Neural Information Processing Systems*, 2014, pp. 2042–2050.

[17] J.-Y. Jiang, F. Chen, Y.-Y. Chen, and W. Wang, "Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking," in *Proceedings of the Annual Conference of North American Chapter of the Association for Computational Linguistics: Human Language Technology*, 2018.

[18] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2014.

[19] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *Proceedings of the AAAI International Conference on Artifical Intelligence*, 2015.

[20] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, 2015.

[21] A. Y. Lin, J. Ford, E. Adar, and B. Hecht, "Vizbywiki: Mining data visualizations from the web to enrich news articles," in *Proceedings of the International Conference on World Wide Web*, 2018, pp. 873–882.

[22] M. Lin, Q. Chen, and S. Yan, "Network in network," in *International Conference on Learning Representations*, 2013.

[23] Y. Lin, B. Yu *et al.*, "Problematizing and addressing the article-as-concept assumption in wikipedia." in *Proceedings of the ACM International Conference on Computer Supported Collaborative Works and Social Computing*, 2017.

[24] E. Meij, K. Balog, and D. Odijk, "Entity linking and retrieval for semantic search." in *Proceedings of the International Conference on Web Search and Data Mining*, 2014.

[25] T. Mikolov, I. Sutskever *et al.*, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013.

[26] Y. Ni, Q. K. Xu *et al.*, "Semantic documents relatedness using concept graph representation," in *Proceedings of the International Conference on Web Search and Data Mining*, 2016.

[27] J. D. Olden, M. K. Joy, and R. G. Death, "An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data," *Ecological Modelling*, vol. 178, 2004.

[28] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the SIGNLL International Conference on Natural Language Learning*, 2009.

[29] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.

[30] D. Rinser, D. Lange, F. Naumann, and G. Weikum, "Cross-lingual entity matching and infobox alignment in Wikipedia," *Information Systems*, vol. 38, no. 6, 2013.

[31] T. Rocktäschel, E. Grefenstette *et al.*, "Reasoning about entailment with neural attention," 2016.

[32] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the ACM SIGIR International Conference on Advances of Information Retrieval*, 2015.

[33] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge." in *Proceedings of the AAAI International Conference on Artifical Intelligence*, 2017.

[34] Z. Sun, W. Hu, and C. Li, "Cross-lingual entity alignment via joint attribute-preserving embedding," in *International Semantic Web Confererence*, 2017.

[35] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2015.

[36] C.-T. Tsai and D. Roth, "Cross-lingual wikification using multilingual embeddings." in *Proceedings of the Annual Conference of North American Chapter of the Association for Computational Linguistics: Human Language Technology*, 2016.

[37] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *International Conference on World Wide Web*, 2012.

[38] Y. Wang, M. Huang, L. Zhao *et al.*, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2016.

[39] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the Annual Conference of North American Chapter of the Association for Computational Linguistics: Human Language Technology*, 2016.

[40] I. Yilmaz and O. Kaynar, "Multiple regression, ann (rbf, mlp) and anfis models for prediction of swell potential of clayey soils," *Expert systems with applications*, vol. 38, no. 5, 2011.

[41] W. Yin and H. Schütze, "Convolutional neural network for paraphrase identification," in *Proceedings of the Annual Conference of North American Chapter of the Association for Computational Linguistics: Human Language Technology*, 2015.

[42] C. Zaniolo, S. Gao, M. Atzori, M. Chen, and J. Gu, "User-friendly temporal queries on historical knowledge bases," *Information and Computation*, vol. 259, pp. 444–459, 2018.

[43] X. Zhang and Y. LeCun, "Text understanding from scratch," *arXiv preprint arXiv:1502.01710*, 2015.

[44] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional lstm with two-dimensional max pooling," in *Proceedings of the International Conference on Computational Linguistics*, 2016.

[45] X. Zhou, X. Wan, and J. Xiao, "Attention-based lstm network for cross-lingual sentiment classification," in *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, 2016.

[46] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, 2009.