RESEARCH ARTICLE

Unsupervised acquisition of idiomatic units of symbolic natural language: An n-gram frequency-based approach for the chunking of news articles and tweets

Dario Borrelli¹*, Gabriela Gongora Svartzman¹, Carlo Lipizzi¹

School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, New Jersey, United States of America

* These authors contributed equally to this work.

* dborrell@stevens.edu

OPEN ACCESS

Citation: Borrelli D, Gongora Svartzman G, Lipizzi C (2020) Unsupervised acquisition of idiomatic units of symbolic natural language: An n-gram frequency-based approach for the chunking of news articles and tweets. PLoS ONE 15(6): e0234214. <https://doi.org/10.1371/journal.pone.0234214>

Editor: Christopher M. Danforth, University of Vermont, UNITED STATES

Received: February 4, 2020

Accepted: May 20, 2020

Published: June 8, 2020

Copyright: © 2020 Borrelli et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Data has been uploaded to the Github Repository: (https://github.com/aritter/twitter_nlp/blob/master/data/annotated/chunk.txt) (<https://github.com/teropa/nlp/tree/master/resources/corpora/conll2000>).

Funding: This paper is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract

Abstract

Symbolic sequential data are produced in huge quantities in numerous contexts, such as text and speech data, biometrics, genomics, financial market indexes, music sheets, and online social media posts. In this paper, an unsupervised approach for the chunking of idiomatic units of sequential text data is presented. Text chunking refers to the task of splitting a string of textual information into non-overlapping groups of related units. This is a fundamental problem in numerous fields where understanding the relation between raw units of symbolic sequential data is relevant. Existing methods are based primarily on supervised and semi-supervised learning approaches; however, in this study, a novel unsupervised approach is proposed based on the existing concept of n-grams, which requires no labeled text as an input. The proposed methodology is applied to two natural language corpora: a Wall Street Journal corpus and a Twitter corpus. In both cases, the corpus length was increased gradually to measure the accuracy with a different number of unitary elements as inputs. Both corpora reveal improvements in accuracy proportional with increases in the number of tokens. For the Twitter corpus, the increase in accuracy follows a linear trend. The results show that the proposed methodology can achieve a higher accuracy with incremental usage. A future study will aim at designing an iterative system for the proposed methodology.

Introduction

Symbolic sequential data are generated every day in huge quantities and include textual data published on the web, data produced by biomedical sensors on human bodies, financial market indexes, and sequences of DNA. Extracting patterns of related units can help those disciplines involved in the rules that generate them. In this study, the focus is on a specific type of raw symbolic sequential data, namely, natural language textual data.

[HQ0034-19-D-0003, TO#0150]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

The unsupervised identification of idioms in raw natural language data requires an unsupervised acquisition of varied elements embedded in a language such as grammar, syntax, semantics, phonetic, phonology, semiotic, and morphology. One system completely capable of achieving this is the human brain. Debate is still ongoing regarding whether a machine can learn such rules from plain text without supervision.

Isolating the syntax would make the learning task easier for a human child, whereas learning the syntax requires information on the semantic, acoustic, and visual elements, all of which contribute to language acquisition [1]. Chomsky [2] introduced the theory of universal grammar, hypothesizing that humans have a language acquisition device (LAD) that allows them to learn the grammar and syntax of any language. According to Chomsky's hypothesis, a human child has the necessary means to speak but does not have yet the experience that allows the child to compile complex sentences together (a so-called "poverty of stimulus").

The process of creating learning models able to parse languages in an unsupervised fashion has received considerable attention from many scholars in the fields of natural language processing (NLP), natural language understanding (NLU), and computational linguistics (CL), among others in which sequential data are relevant. Although syntax and grammar induction are a simplification of the human process when acquiring a language, the widespread approach of isolating the syntax is based on representational modularity theory, which states that the human mind encodes information in separate layers and formats, and that syntax is one of these layers [3]. By contrast, it is unclear whether additional layers of semantics and phonology [4, 5] can help improve the text chunking process for a machine.

Most supervised approaches for learning syntax use hand-parsed sentences on large treebanks, which are corpora analyzed and annotated with the help of domain experts. The NLP community has recently been making extensive use of an unsupervised method for generating a distributional representation of text units based on an artificial neural network, called text embedding. The units applied by the embedding models can vary from the scale (see the article by Febres et al. [6] for the concept of scale) of characters, to the scale of words, up to the paragraph and document levels. Many NLP tasks have been improved with the application of these new vectorization techniques, including text chunking [7–10]. Parse trees, syntax annotations, and part-of-speech (PoS) tagging are not necessary for training these embedding models. In fact, such models require a set of textual units as an input without having a specified syntactic role that each textual unit must have. By contrast, most of the existing approaches for shallow parsing are based on syntax annotations. The use of syntactical annotated treebanks has been the primary option of many supervised text chunking methods, typically demonstrating a high accuracy. However, the accuracy of such supervised-learning-based chunkers degrades widely when applied to new texts, new genres, and new languages [11]. The unsupervised learning of natural language is a difficult but basic research problem, as highlighted by Klein and Manning [11]. After numerous years of effort, the accuracy of the results is still unacceptably low when compared against the high accuracy of semi-supervised and supervised approaches. However, the supervision provided to these models leads to a lack of generalization, particularly among the contexts in which natural language deviates from traditional syntax rules (e.g., social media). Unlike other works that intend to induce a tree structure that governs natural language, this work generalizes to the simplest detection of sequences of symbolic units that occur with a certain frequency. In this way, the problem of domain adaptation is addressed because it is possible to extend the method to other diverse sequential data. This work illustrates an intuitive unsupervised approach evaluated on two different types of corpora: news and social media text.

The main contribution of this research is the proposed methodology. This methodology is based on n-gram theory to create text chunking classification without any tags, i.e.,

unsupervised learning. A chunk is a unit of text that refers to a specific syntactical phrase, which can be a noun-phrase or a verb-phrase. The proposed methodology has the flexibility of accepting text in different languages and from different sources, without relying on any tags or pre-classification. The motivation in creating this methodology lies in the possibility of improving any NLP task that currently requires labels to compute, even providing the possibility of a real-time text analysis. This can have an impact on understanding one's perception from a text, creating narratives from text, pursuing security against cyber-threats, and even improving artificial natural language systems designed to have a positive impact on society.

The remainder of this paper is organized into five main sections. The current section introduces the problem, the proposed approach, and the motivation behind it. The next section, describes previous studies on semi-supervised, supervised, and unsupervised approaches. The following section describes in detail the methodology proposed for implementing and evaluating unsupervised text chunking. The methodology applied to two different corpora is then described by measuring and comparing their accuracy and performance. These experimental results are then described in the next section. Finally, some concluding remarks are provided along with limitations and areas of future study.

Related studies

Supervised and semi-supervised approaches

Supervised or semi-supervised text chunking models typically rely on a gold standard annotated corpus used for training a machine learning or deep learning model. The Conference on Computational Natural Language Learning (CoNLL) created a shared task specifically for text chunking [12]. Since 2000, the CoNLL-2000 corpus has been the default standard for benchmarking between different text chunking methodologies.

The CoNLL-2000 corpus consists of 8,936 training sentences and 893 test sentences extracted from Wall Street Journal articles from the English Treebank. The corpus has been manually annotated following the Begin-Inside-Outside (BIO) labeling standard, which assigns a label to each token of the corpus. Distinctively, in any given sentence, single-word-tokens are extracted and annotated with one of the BIO labels. The "B" in BIO demonstrates that the token is the first word of the chunk, "I" indicates that the token is inside (or at the end) of a chunk, and "O" shows that the token is outside any chunk. Having an annotated corpus leads intuitively to a supervised learning model approach.

Numerous authors have adopted supervised text chunking using Hidden Markov Models (HMM) for predicting PoS tags and forming chunks accordingly to syntactic rules [13–16]. Other groups have approached the same problem using conditional random fields (CRFs) [17–19]. Owing to their specific characteristics, both methods have been extensively adopted for studying different sequential data. A recent research direction has applied supervised text chunking tasks using artificial neural network architectures. Zhai et al. [20] showed successful results when applying a recurrent neural network (RNN), long short-term memory (LSTM), and their variations.

A significant number of studies have adopted pre-trained word embeddings for improving the results of the above-mentioned techniques [21–23]. Pre-trained embeddings are also being used in this field, including bidirectional encoder representations from transformers (BERT) applied by Devlin et al. [24] and embeddings from language models (ELMo) developed by Peters et al. [25]. Both BERT and ELMo have been evaluated for their syntactic recognition capabilities, which can improve the supervised text chunking [26].

In general, a global ranking is assumed for the quality of the available pre-trained embeddings. However, Faruqui et al. [27] and Schnabel et al. [28] found that the accuracy of text

Table 1. Approaches from the literature, among which a few are unsupervised.

Authors	Learning Paradigm			Embeddings	Language	Corpus
	Supervised	Semi-supervised	Unsupervised			
[20, 22, 23, 31, 43–45]	✓			✓	English	WSJ news
[37]	✓			✓	English	Biomedical corpus
[41]	✓				Others	Software code
[46]	✓				Others	Thai corpus
[40]	✓				Others	Arabic treebank
[32, 47]		✓		✓	English	WSJ news
[38]			✓		Others	Chinese treebank
[48–50]			✓		English	News

<https://doi.org/10.1371/journal.pone.0234214.t001>

chunking methods using pre-trained embeddings does not necessarily reflect the same global rankings. For this reason, different combinations [29, 30] of pre-trained embeddings have been tested for different chunking methods, i.e., from the scale of characters [31] to the scale of words using supervised and semi-supervised approaches [32]. Supervised shallow parsers have become publicly available using software and programming language library solutions [33–35]. Abnik et al. [35] made their chunker available in a Python library called Flair. Other NLP libraries (SpaCy and Natural Language Tool Kit) also provide users with supervised noun-phrase chunkers trained on the CoNLL-2000 corpus. Supervision limits the parsing model to the available annotated corpora, and therefore a supervised approach for text chunking has been used for context biased text with domain- and language-specific annotated corpora: for instance, chunkers for the bio-medical and medical fields [36], chunkers for software engineering [37], Chinese, Thai, and Arabic language chunkers [38–40], and social media text chunkers [41–43]. Table 1 groups the text chunking methods found in the literature, whereas Table 2 shows an overview of supervised approaches for text chunking with their respective performances on specific annotated datasets.

Unsupervised approaches to text chunking

Several studies on NLP and CL have focused on unsupervised machine learning approaches for learning languages. Some studies have generalized the learning tasks to different natural languages, for example, biological sequences, text or speech [52], and music [53, 54], among

Table 2. Remarkable performances of supervised methods.

Authors	Data set	Domain	$F_{\beta=1}$	Method
[32]	CoNLL-2000	News	96.98%	GloVe Embeddings + biLSTM + CVT
[31]	CoNLL-2000	News	96.72%	LSTM + Character Embeddings
[51]	CoNLL-2000	News	96.49%	Hidden Markov Model
[22]	CoNLL-2000	News	96.32%	Bi-Language Model
[20]	CoNLL-2000	News	95.86%	LSTM Bi-directional & Pointer Network
[36]	GENIA Corpus	Biomedical corpus	95.70%	Gate Chunker
[47]	CoNLL-2000	News	93.88%	Word embeddings + POS tags supervision
[43]	CoNLL-2000	News	93.50%	RNN + LSTM
[51]	Penn Arabic Treebank	News	93.48%	CRF
[44]	CoNLL-2000	News	93.48%	Support Vector Machine
[23]	Chinese Corpus	Mixed	91.76%	Character Recognition, CRF & RNN
[41]	Alan Ritter	Twitter	86.7%	TwitterNLP

<https://doi.org/10.1371/journal.pone.0234214.t002>

others [55], which are represented through sequential symbols [56]. Seginer [57] introduced a model called common cover links (CCL), which learns dependencies in a text (in a tree form) without any previous annotation. Subsequently, Ponvert [48] explored this method and discovered that the performance obtained from unsupervised parsing with the CCL significantly depended on the identification of low-level constituents, i.e., words that compose the lowest leaves of a parse tree.

The identification of low-level constituents is considered a “partial” parsing, because the internal role of each word in a chunk is not really known or taken into consideration. Hence, unsupervised partial parsing is a sub-problem of unsupervised parsing. To this extent, unsupervised partial parsing has been defined as the “unsupervised version of text chunking” by Ponvert et al. [58], who also presented a simple unsupervised bi-gram model comparable to a CCL unsupervised parser. This is an important result for two main reasons: It demonstrates that low-level constituents are the elements that matter for full parsing, and it proves that the type of chunk representations used are closer to supervised text chunking outputs.

A reputable unsupervised grammar induction method was proposed by Klein and Manning [59] and later improved upon by Headden et al. [60] and Cohen [61]. Each of these methods assumes the PoS to be known, as derived from the gold standard annotation. Seginer [57], followed by Ponvert et al. [49], provided grammar induction models for learning from plain text alone without any given manual annotation. The goal of text chunking is to identify the low-level constituents in a text without providing the tree structure of sentences that are typical of dependency or constituency parsing. This issue has been addressed in the past using traditional grammar induction and unsupervised methods. The goal of unsupervised text chunking is the same, with the exception that chunks must be learned without providing any form of supervision (e.g., treebank annotations or human feedback).

Abney [62] highlights the differences between low-level constituents and chunks. A low-level constituent (called a “clump” by Ponvert et al. [58]) has two main differences with a chunk: the labeling and the additional components found in the chunks. In 2000, a shared task on text chunking was introduced by Tjong at al. [12]. This task solved the chunking problem with the help of a dedicated labeled dataset. It is still used as a benchmark corpus, and a high accuracy has been obtained by applying different supervised and semi-supervised learning methods [31, 32]. One of the main advantages of unsupervised text chunking is the independence from the corpus to the chunk-specific domain. In fact, an unsupervised method does not require the manual construction of domain-specific annotations. Because they must be performed by domain experts, the placement of annotations and the building of gold-standard treebanks for a specific domain are time-consuming and labor intensive.

As previously mentioned, it was proven by Mikolov et al. [63] that the performance of supervised methods, which rely on gold annotated treebanks, decays dramatically when applied to other domains or other languages. Mikolov et al. [63] also mentioned that the generation of distributed representations of textual units would be adopted in NLP owing to the improvements they provide to different NLP tasks (e.g., the text classification task by Jang et al. [64]).

Such techniques, including Word2Vec, acquire words or chunks of words as an input and generate vectors as an output. The input units do not need to be tagged with the PoS tags or any other annotation that follows syntax and grammar. However, the input units must be meaningful to obtain a significant vector representation. Hence, embedding models require unsupervised methods for text chunking to produce chunks ready to be processed. An important consideration highlighted by Lightfoot [65] and Cocho et al. [66] was that language changes over time and so do the uses of the same languages. Chunks obtained using a supervised method with an annotated corpus produced 20 years ago may not work properly today.

Basic pre-processing (Fig 1a)

This first step is made up of two sub-steps:

1. *Lower casing*—Every character has a lower-case version in the plain text to avoid potential duplicate strings, for example, “Machine” and “machine.” The lower-case text is used as an input to a function separating the punctuation characters from neighboring words. For instance, “this is a pen.” is transformed into “this is a pen.” by adding a space between “pen” and the period. This is executed to avoid further duplicates that may reduce the quality of the final chunks.
2. *Punctuation and spacing*—This sub-step simply implies removing any punctuation symbols (e.g., “?” and “;”), special characters, and spacing that can cause problems when analyzing the text.

Detection of frequent chunks (Fig 1b)

This step involves calculating the frequency of repeated n -grams for $n = 2$ until $n = \max$. The maximum value ($\max$) of n is the length of the longest sequence of tokens repeated at least once in the plain pre-processed text. For every n , the chunks with a frequency of greater than 1 are stored in a separate set. The output is a set containing all candidate chunks. An example of this list is as follows: {“machine learning,” “machine learning is,” “machine learning algorithms,” “learning algorithms,” . . .}. This list of candidate chunks is noisy. In fact, it contains 2-grams and 3-grams that present a common part (e.g., “machine learning,” “machine learning is,” “machine learning algorithms”). Preliminary parameters are computed while scanning the input corpus to explore patterns of co-occurrences of unitary tokens. These parameters are required for a calculation of the frequencies of co-occurring n -grams. In turn, these frequencies are the basis for chunking any corpus. The four parameters can be described as follows:

1. *Total non-unique number of tokens in the corpus (T)*. For example, considering the small corpus, “text chunking is an important task in NLP,” the total non-unique number of tokens is eight. This value has been calculated by spacing the punctuation, splitting the corpus into tokens using spaces, and counting their resulting number.
2. *The total number of repeated n -grams* is a relevant and related metric. It counts the number of repeated n -grams, i.e., the total number of n -grams with a frequency of greater than 1. In the same example from the first metric, the total number of repeated n -grams is equal to zero because none of the possible n -gram are repeated within the corpus. If the corpus is changed into “Text chunking is an important task in NLP. Text chunking is also known as shallow parsing,” the number of repeated 2-grams is equal to 1 because there is only one n -gram with a frequency of greater than 1. “Text chunking” is repeated twice. This can be calculated through the following procedure. First, the total number of n -grams (η) can be represented using Eq 1:

$$\eta = \left[\sum_{n=2}^{\max} (T + 1 - n) \right] + T \quad (1)$$

The set τ of all repeated n -grams can then be written as follows:

$$\tau = \{x_1, x_2, \dots, x_\eta\} \quad (2)$$

where x_i is a generic i^{th} n-gram, and $|\tau| = \eta$. By calculating the frequencies of each n-gram in τ , it is possible to obtain the following set:

$$\mathcal{R} = \{x_i \in \tau : F_{x_i} > 1\} \subseteq \tau \quad (3)$$

where F_{x_i} is the frequency of a generic i^{th} n-gram x_i . The cardinality of the set $\mathcal{R}$ provides the total number of repeated n-grams (ρ):

$$\rho = |\mathcal{R}| \quad (4)$$

3. *The sum of all n-gram frequencies that repeatedly appear when pre-processed.* In the example “text chunking is an important task in NLP. Text chunking is also known as shallow parsing.” This metric, calculated for 2-grams, is equal to 2 because the sum of the repeated bi-gram frequencies is equal to 2. This metric can be formalized through Eq 5:

$$\varphi = \sum_{x_i \in \mathcal{R}} F_{x_i} \quad (5)$$

4. *The total number of unique tokens.* This metric is equivalent to the total number of unique tokens in the plain pre-processed text, i.e., the total non-unique number of tokens when there are no repeated tokens in the corpus. Because the proposed methodology is dependent on the frequencies of repeated patterns, this metric is an indication of how much the corpus has n-gram repetitions, hence, potential chunks.

Chunks cleaning (Fig 1c)

This step refers to the automatic cleaning of the list created in the previous step from the incorrect chunks (e.g., “machine learning is”). Consider two candidate chunks C_i, C_j . Both can be represented as an ordered set of tokens described by Eq 6. The intersection between the two sets of tokens is then calculated to check if the following equivalence is true through Eq 7. In the affirmative case, C_i will be excluded from the final list of candidate chunks only if $T_{j|C_j|}$ is one of the most-frequent 1-grams in the initial corpus, and if it has been typically found to be an English stop word or a punctuation character for an English corpus.

$$C_i = (T_{i1}, \dots, T_{i|C_i|}), \quad C_j = (T_{j1}, \dots, T_{j|C_j|}) \quad (6)$$

$$C_i \cap C_j = C_i - (T_{j|C_j|}) \quad (7)$$

where $|C_i|, |C_j|$ represent the cardinality of the set C_i and the cardinality of the set C_j , respectively, with $|C_i| > |C_j|$.

Chunks replacement (Fig 1d)

This step uses the cleaned list (from the previous step) to replace the final chunks into the initial corpus. Fig 2 shows the overall chunking process with a visual representation. In other words, the chunk replacement step takes each chunk from the list of chunks from the previous cleaning step and replaces it within the raw corpus using a replacement of spaces (“ ”) with any other non-alphanumeric character not present in the raw corpus (for instance, the underscore symbol “_”). For example, the 3-gram “the current account” is replaced by “the_current_account.”

Pseudo-labeling (Fig 1e)

This step uses chunks for labeling and comparing them with those in the annotated corpus.

Fig 2. Chunking. Visual representation of the chunking process: (a) identification of repeated n-grams and (b) max frequency values over n .

<https://doi.org/10.1371/journal.pone.0234214.g002>

The proposed methodology does not directly provide labels to be compared with those in the annotated corpus for evaluating the chunking results. For this reason, pseudo-labels are generated. Pseudo-labels are created for each token by assigning one of the BIO labels. For example, the tokens of the chunk “the current account deficit” will have the following pseudo-labels: [the]B [current]I [account]I [deficit]I, or the chunk “will narrow” will have the following labels per token [will]B [narrow]I. This is useful for evaluating the proposed method with annotated corpora that use BIO labels, such as the CoNLL-2000 Corpus [12] and the Alan Ritter Twitter Corpus [67]. Examples of these pseudo-labels are shown in Table 3, whereas examples of BIO labels are shown in Table 4.

As previously mentioned, this method aims to obtain text chunks without providing an indication of the PoS. Hence, a pseudo-label for “B,” “I,” and “O” can be assigned, although the internal role of each token (PoS annotation) is not provided. Both corpora used for the evaluation (Penn Treebank CoNLL-2000 for newswire articles and Alan Ritter Corpus for Twitter text data) provide a BIO labeling schema.

Evaluation (Fig 1f)

This last step provides evaluation metrics, such as the $F_{\beta=1}$ score, to assess the accuracy of the method in different types of corpora. Following the traditional choice of evaluation for a text chunking task, the accuracy is calculated by comparing the BIO labels of the predicted chunks (pseudo-labels in this case) with the BIO labels of the original chunks using the $F_{\beta=1}$ score [68, 69]:

$$F_{\beta} = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad (8)$$

Table 3. Unsupervised paradigm for the attribution of pseudo-labels.

Unsupervised paradigm	
input	He reckons the current account deficit
output	[He] [reckons] [the current account deficit]
pseudo – labels	[He]B [reckons]B [the]B [current]B [account]I [deficit]I

The PoS of each token is ignored in the unsupervised chunker.

<https://doi.org/10.1371/journal.pone.0234214.t003>

Table 4. Traditional paradigm for the attribution of BIO labels.

Supervised and Semi-supervised paradigm	
<i>input</i>	He reckons the current account deficit
<i>output</i>	[He]B _{NP} [reckons]B _{VP} [the]B _{NP} [current]I _{NP} [account]I _{NP} [deficit]I _{NP}
<i>BIO labels</i>	[He]B _{NP} [reckons]B _{VP} [the]B _{NP} [current]I _{NP} [account]I _{NP} [deficit]I _{NP}

The output comes with BIO-labels in both supervised and semi-supervised chunkers.

<https://doi.org/10.1371/journal.pone.0234214.t004>

$$F_{\beta=1} = 2 \cdot \frac{P \cdot R}{P + R} \quad (9)$$

$$\text{Precision } (P) = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \quad (10)$$

$$\text{Recall } (R) = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \quad (11)$$

where *precision* P is the percentage of chunks that are correct and *recall* R is the percentage of chunks that were found by the unsupervised text chunker; in addition, *true positives* is the number of tokens that have been correctly predicted as “B” (beginning of a chunk), *false positives* is the number of tokens that are incorrectly predicted as the begin of a chunk “B,” and *false negatives* indicates the number of tokens that are not predicted as “B” although they actually are at the begin of a chunk. It is worth noting that the proposed unsupervised approach does not naturally provide labels (predictions) to calculate F_{β} , P , and R . Nevertheless, to conduct a comparison with existing approaches for text chunking, which are mostly based on labeled datasets (similar to the annotated corpora used in this study), pseudo-labels are generated, as described previously in this section. Hence, the predictions are the pseudo-labels and the true labels are the annotated labels.

Results and discussion

Previous studies evaluating supervised learning methods have typically shown a high accuracy. These results have generally been obtained by training and testing models using an annotated treebank. However, this high accuracy degrades when the same model is applied to text from a different domain. This has increased and directed the attention toward unsupervised approaches. In this study, textual chunks were obtained using a completely unsupervised approach based on the frequency of n-grams combined with automated pre-processing and cleaning steps.

The frequencies of different n-grams are calculated by varying the corpus length expressed as the number of tokens. Fig 3 shows an example in which the length of the corpus has been varied (with six different lengths). The length is indicated at the top-left of each plot. The larger graph (with blue data points) shows the value of n along the x-axis, and the n value and maximum frequency of the n-grams at each n are shown along the y-axis. When the length of the corpus increases, the value of n at the maximum frequency (where $F > 1$) also increases. Beyond this value of n , there is a frequency convergence at 1, meaning that there are no repeated n-grams after that value.

In the smaller graph (upper-right corner), the numbers of unique n-grams are plotted as a function of n . It is possible to see how this number decays when n increases in a small corpus.

Fig 3. Maximum frequency of chunks varying the corpus length. Exploration of dependencies between the maximum frequency of n-grams and the length of the input corpus in terms of the number of tokens.

<https://doi.org/10.1371/journal.pone.0234214.g003>

This is because, with a large n and a relatively small corpus, many n-grams cannot be generated, i.e., there is a limited number of available tokens. By increasing the size of the corpus, it can be seen that the value stabilizes and sets on a certain number because n-grams are not repeatedly found above a specific value of n , as shown by the yellow dot.

This initial exploration helps calculate the maximum value of n and provides insight into the type of input corpus required to build the n-grams. An unsupervised approach has been applied to two different corpora: a newswire corpus and a social media text corpus. The choice of these two different datasets allows the proposed approach to be compared to two different types of textual styles: news and social media. The first is a more conventional type of text, whereas the second is an informal type of text in which grammatical rules may be broken. The newswire corpus is part of the English Penn Treebank II (sections 15-16-17-18-20) consisting of Wall Street Journal news [12, 70]. It has been extensively used for supervised text chunking in the literature, as previously mentioned in the introduction. The social media corpus is the Twitter corpus proposed by Alan Ritter [67].

Fig 4. Linear fit of the two sets of data points. Regression line for both analyzed corpora, with the WSJ corpus on the left and Twitter corpus on the right.

<https://doi.org/10.1371/journal.pone.0234214.g004>

Fig 4 shows two graphs in which the accuracy of the proposed method (measured using $F_{\beta=1}$) on the two different corpora is plotted over the number of non-unique tokens. The left side contains the graph for the WSJ corpus, whereas the right side contains the results for the Twitter corpus. As can be seen, with an increasing number of tokens, the accuracy in both corpora increases, as might be expected, because if more data are available more frequent chunks are considered to be counted. However, a striking difference was noted between the two increasing rates. The accuracy ($F_{\beta=1}$ score) with the Twitter corpus increases with a linear tendency. By contrast, the WSJ corpus shows a reluctance in increasing linearly after a certain number of tokens. In fact, the left graph of Fig 5 shows a point after which the precision (for the WSJ corpus) reaches a plateau with an increase in the number of tokens. This behavior was also registered with a reduced WSJ corpus using a comparable number of tokens as the Twitter Corpus length (central graph in Fig 5).

To properly assess the linearity of the graphs, a linear model was fitted to the data points. Using this approach, the accuracy on the Twitter corpus showed a more linear increase compared to that of the WSJ corpus, which showed a logarithmic tendency. These results suggest that the proposed unsupervised method for text chunking may provide better results when applied to Twitter data compared with the newswire corpora. The accuracy of the chunker on the newswire corpus had a logarithmic tendency rather than a linear tendency, providing a

Fig 5. Accuracy, precision, recall. The $F_{\beta=1}$ score, precision, and recall as a function of the corpus length. In this graph, the WSJ corpus length has also been reduced to the same dimension as the Twitter Corpus.

<https://doi.org/10.1371/journal.pone.0234214.g005>

Table 5. Comparison of accuracy for the highest number of tokens available.

Corpus	$F_{\beta=1}$	P	R
WSJ—Penn English Treebank	.64729	.67278	.68463
Twitter Corpus (Alan Ritter)	.59762	.68037	.62291

<https://doi.org/10.1371/journal.pone.0234214.t005>

risk of an asymptotic convergence to a higher accuracy. This may suggest an inefficient behavior if the newswire corpus is used recursively. Both graphs of Fig 4 show an increase in accuracy as the size of the corpus increases. For the WSJ corpus, an initial rapid and regular increase in accuracy can be seen. For the Twitter corpus, the method achieves a more fluctuating accuracy at the beginning and then grows uniformly afterward. Fig 4 shows a linear generalization of the accuracy trend. It can be seen that on the Twitter corpus the unsupervised method achieves a linear growth in performance with the number of input tokens, whereas for the WSJ corpus, a linear fitting seems to be inadequate. This corpus presents a logarithmic trend with a tendency to converge toward a higher accuracy. This result may favor unsupervised text chunking models for social media texts in this specific case of Twitter, in which the language often does not strictly follow pre-defined grammar/syntax rules. The proposed method based on n-gram frequencies is completely independent from the syntax and grammar rules that govern the input language. Table 5 summarizes the best results obtained with the proposed method on these two different corpora. In addition, Table 6 shows the differences between state-of-the-art approaches that apply these two standard annotated data-sets for text chunking. The focus is mainly on two aspects: (1) the substantial use of the WSJ newswire corpus (CoNLL-2000), and (2) the direction of the approaches presented in the literature toward methodologies that include at least semi-supervision (S-SUP) during the learning process. Consequently, the $F_{\beta=1}$ score obtained by the semi-supervised methods (S-SUP) is quite high (see Table 2). However, among an extremely small number of unsupervised methods (UNS), the corpus developed by Ponvert et al. achieved an $F_{\beta=1}$ score and precision P slightly higher than those obtained in the present study. By contrast, in their paper [49], the recall turns out to be slightly lower ($R = .639$) than that found with our approach ($R = .684$). Despite this, the unsupervised method proposed herein (proposed UNS) is the only method that has been applied to Twitter (among those testing the method on these two standard corpora). This represents a novelty, and highlights a gap, in the literature. It is important to note that the results of these annotated corpora are not yet competitive with the accuracy of existing supervised learning studies. Unsupervised learning of natural language is still a challenge for NLP and other fields where the aim is to learn the rules underlining sequential data in an unsupervised fashion. In addition, the application of such approaches to sequential information differing from written natural language can be a challenge. For instance, as stated by Febres et al. [54],

Table 6. Comparison of $F_{\beta=1}$ with state-of-the-art methods.

	WSJ Corpus		Twitter Corpus	
	S-SUP	UNS	S-SUP	UNS
Peters et al. [22]	.963	-	-	-
Clark et al. [32]	.967	-	-	-
Ritter et al. [67]	.854	-	.867	-
Ponvert et al. [49, 58]	-	.698	-	-
Proposed UNS	-	.647	-	.597

<https://doi.org/10.1371/journal.pone.0234214.t006>

whereas a space in natural language (i.e., the English language) is used as delimiter, this may not be true with other symbolic data. In such situations, the adoption of algorithms such as the fundamental scale algorithm [56] based on a minimization of the overall entropy can be a worthwhile direction. Moreover, incorporating symbolic diversity and symbolic entropy in unsupervised approaches, as with the study by Febres et al. [53], can be worth exploring when comparing different symbolic sequential data.

Conclusion

In this study, an unsupervised approach to text chunking was introduced as one of the basic tasks in NLP. A methodology for extracting chunks in an unsupervised fashion was presented as an alternative solution to applications in which a labeled treebank is unavailable.

In this study, the presented methodology for unsupervised text chunking was compared using two different corpora: a newswire corpus composed of Wall Street Journal articles and a Twitter corpus of random tweets. Both corpora have manual chunk annotations identified by human experts. These corpora are typically used for evaluating supervised and semi-supervised text chunkers. This provides a means to evaluating the proposed unsupervised method using accuracy metrics. In fact, a proper pseudo-labeling step was conducted, converting an unlabeled output into a labeled one. To measure the accuracy with the $F_{\beta=1}$ score, chunks are converted in such a way that pseudo-labels assigned to each token can be compared with the original labels of tokens in both corpora.

The accuracy of the unsupervised chunking method was compared between the news-wire corpus and the Twitter corpus. For the Twitter corpus, the $F_{\beta=1}$ score followed a linear trend with an increase the number of tokens (the length of the corpus). The precision (P) and recall (R) also showed a similar dynamic. The news-wire corpus presented the opposite behavior, the accuracy of which did not follow the same tendency. The annotations upon which the accuracy was measured are based on a syntax schema; hence, it can be interpreted that the WSJ corpus conveys more complex syntactical rules than the Twitter corpus. The limited number of characters per tweet (140 characters) and the informal way in which people communicate through online posts in comparison to the language used in WSJ news articles may play a role in favoring an unsupervised chunker for Twitter.

The proposed chunker is based on a simple mechanism of counting repeated occurrences of n -grams, which is an increasingly more effective mechanism when more data become available. The log-linear tendency found by Pitler et al. [71] on their web news corpus is also evident in the WSJ corpus. However, it is worth noting that the problem of the asymptotic tendency of a log-linear function of the accuracy (evident in the WSJ corpus) can lead to the consequential risk of convergence.

The performance of this method was measured based on the F_1 score. The accuracy of this method for two different types of text was shown to increase gradually when increasing the number of tokens as an input. This approach has demonstrated the potential to be more accurate with recursive usage. The more un-labeled data that are provided, the greater the chances of capturing new repeated n -grams, which are chunk candidates. This is in line with the recurrent idea that “more data are better data,” indicating that the more chunks the algorithm can obtain for its training the better it will perform. Nonetheless, the quality of these chunks and the type of text where the chunks come from are as important for acquiring more accurate results in unsupervised chunking methodologies. For instance, in the study by Pitler et al. [71], a web-scale n -gram model proved that the accuracy of noun phrase identification increase the log-linearly when more data are available. The datasets used for the evaluations are an important aspect to consider. For instance, the results reported in this paper refer to datasets

designed to apply supervised methodologies with the support of different labels. In the presented case, during the application of the algorithm, the labels were ignored. However, these labels were then used in the evaluation phase. Hence, the application of this method can be extended to any type of sequential data. By contrast, the evaluation phase presented here requires the presence of annotations in the corpus, which can be non-trivial in finding non-textual sequential data. Furthermore, a type of labeling (and pseudo-labeling) schema differing from the conventional labels of the treebanks may be required.

Future studies will test the method on other types of sequential data. It is worth noting that human annotations may contain biases. Consequentially, measuring the accuracy on these labels can lead to biased conclusions. This bias is intrinsic in the measurement error and subjectivity of human annotation. However, attempts have been made to minimize this bias by using gold treebanks, in which annotations have been made by domain experts. The presence of new languages that are created by online users may not reflect the standards used by domain experts. In the unsupervised approach presented herein, the learning task for a machine does not occur with the supervision of a human expert, but with a repetition-based approach. This approach, for certain aspects, reflects a similarity with the learning process of new languages by children. Like Skinner's empiricist theory of language acquisition, other languages, encoded with sequential data, can be learned by a machine with a repetition-based strategy. The lack of experimentally measured precision in the results may be metaphorically linked to the absence, for a machine, of the innate faculty of the human brain in the acquisition of language, as theorized by Chomsky.

Acknowledgments

Special thanks to Rushabh Barbhaya, Pedro Veronezi, and Paola De Simone who helped during the initial stages of this research.

Author Contributions

Conceptualization: Dario Borrelli, Gabriela Gongora Svartzman, Carlo Lipizzi.

Data curation: Dario Borrelli.

Formal analysis: Dario Borrelli.

Funding acquisition: Carlo Lipizzi.

Investigation: Dario Borrelli.

Methodology: Dario Borrelli.

Project administration: Carlo Lipizzi.

Resources: Dario Borrelli, Carlo Lipizzi.

Software: Dario Borrelli, Gabriela Gongora Svartzman, Carlo Lipizzi.

Supervision: Carlo Lipizzi.

Validation: Dario Borrelli.

Visualization: Dario Borrelli, Gabriela Gongora Svartzman.

Writing – original draft: Dario Borrelli, Gabriela Gongora Svartzman.

Writing – review & editing: Dario Borrelli, Gabriela Gongora Svartzman.

References

1. Roark B, Pate JK, Goldwater S. Unsupervised Dependency Parsing with Acoustic Cues. *Transactions of the Association for Computational Linguistics*, 1 (2013) 63–74. 2013.
2. Chomsky N. *Aspects of the Theory of Syntax* [Internet]. MIT Press. 1965.
3. Jackendoff R. Fodorian Modularity and Representational Modularity. *Language and the Brain*. 2007. pp. 3–30.
4. Pate JK, Goldwater S. Unsupervised syntactic chunking with acoustic cues: computational models for prosodic bootstrapping. *Proc 2nd Work Cogn Model Comput Linguist*.
5. Kamper H, Jansen A, Goldwater S. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Trans Audio Speech Lang Process*. 2016; 24: 669–679. <https://doi.org/10.1109/TASLP.2016.2517567>
6. Febres G. A Proposal about the Meaning of Scale, Scope and Resolution in the Context of the Information Interpretation Process. *Axioms*. 2018; 7: 11. <https://doi.org/10.3390/axioms7010011>
7. Kusner M, Sun Y, Kolkin N, Weinberger K. From word embeddings to document distances. In *International conference on machine learning 2015 Jun 1* (pp. 957–966).
8. Schnabel T, Labutov I, Mimno D, Joachims T. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing 2015 Sep* (pp. 298–307).
9. Kimothi D, Biyani P, Hogan JM, Soni A, Kelly W (2020) Learning supervised embeddings for large scale sequence comparisons. *PLoS ONE* 15(3): e0216636. <https://doi.org/10.1371/journal.pone.0216636> PMID: 32168338
10. Wang Y, Liu Z, Sun M (2015) Incorporating Linguistic Knowledge for Learning Distributed Word Representations. *PLoS ONE* 10(4): e0118437. <https://doi.org/10.1371/journal.pone.0118437> PMID: 25874581
11. Klein D. *The unsupervised learning of natural language structure*. PhD Thesis. 2005.
12. Tjong EF, Sang K, Buchholz S. Introduction to the CoNLL-2000 Shared Task: Chunking [Internet]. 2000.
13. Ratnaparkhi A. A Maximum Entropy Model for Part-of-Speech Tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1996. pp. 133–142.
14. Orosz G, Novák A. PurePos 2.0: A hybrid tool for morphological disambiguation. *International Conference Recent Advances in Natural Language Processing, RANLP*. 2013.
15. Orosz G, Novák A. PurePos: An Open Source Morphological Disambiguator. 2012. pp. 53–63.
16. Brants T. TnT-A Statistical Part-of-Speech Tagger. 2000.
17. Sha F, Pereira F. Shallow parsing with conditional random fields. 2003.
18. Müller T, Schmid H, Schütze H. Efficient Higher-Order CRFs for Morphological Tagging. *Proc 2013 Conf Empir Methods Nat Lang Process*. 2013; 322–332.
19. Muis AO, Lu W. Weak Semi-Markov CRFs for Noun Phrase Chunking in Informal Text. 2016; 714–719.
20. Zhai F, Potdar S, Xiang B, Zhou B. Neural Models for Sequence Chunking. 2017.
21. Alemi AA, Ginsparg P. Text Segmentation based on Semantic Word Embeddings [Internet]. 2015.
22. Peters M, Ammar W, Bhagavatula C, Power R. Semi-supervised sequence tagging with bidirectional language models [Internet]. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.
23. Lyu C, Chen B, Ji D. Multi-perspective embeddings for chinese chunking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018. pp. 570–579.
24. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.
25. Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018. pp. 2227–2237.
26. Jawahar G, Sagot B, Seddah D, Benoit GJ, Sagot B. What does BERT learn about the structure of language? [Internet]. 2019.
27. Faruqui M, Tsvetkov Y, Rastogi P, Dyer C. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. 2016. pp. 30–35.

28. Schnabel T, Labutov I, Mimno D, Joachims T. Evaluation methods for unsupervised word embeddings. *Association for Computational Linguistics*; 2015. pp. 298–307.
29. Ghannay S, Favre B, Esteve Y, Camelin N. Word embedding evaluation and combination. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. 2016. pp. 300–305.
30. Lee Y-Y, Ke H, Huang H-H, Chen H-H. Combining word embedding and lexical database for semantic relatedness measurement. *Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee*; 2016. pp. 73–74.
31. Akbik, Alan, Duncan Blythe, Roland V. Contextual string embeddings for sequence labeling [Internet]. *Proceedings of the 27th International Conference on Computational Linguistics*. 2018.
32. Clark K, Luong M-T, Manning CD, Le Q. Semi-Supervised Sequence Modeling with Cross-View Training. 2019. pp. 1914–1925.
33. Loper E, Bird S. NLTK: The Natural Language Toolkit [Internet]. 2002.
34. Honnibal M, Johnson M. An Improved Non-monotonic Transition System for Dependency Parsing. *Association for Computational Linguistics*; 2015.
35. Akbik A, Bergmann T, Blythe D, Rasul K, Schweter S, Vollgraf R. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP [Internet]. 2019.
36. Kang N, van Mulligen EM, Kors JA. Comparing and combining chunkers of biomedical text. *J Biomed Inform*. 2011; 44: 354–360. <https://doi.org/10.1016/j.jbi.2010.10.005>
37. Gupta S, Malik S, Pollock L, Vijay-Shanker K. Part-of-speech tagging of program identifiers for improved text-based software engineering tools. *IEEE International Conference on Program Comprehension*. 2013.
38. Zhu L, Wong DF, Chao LS. Unsupervised Chunking Based on Graph Propagation from Bilingual Corpus. *Sci World J*. 2014.
39. Zhou J, Qu W, Zhang F. Exploiting Chunk-level Features to Improve Phrase Chunking [Internet]. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics*; 2012.
40. Khoufi N, Aloulou C, Belguith LH. Chunking Arabic texts using Conditional Random Fields. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*. 2014. pp. 428–432.
41. Derczynski L, Ritter A, Clark S, Bontcheva K. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data [Internet]. *Proceedings of the Recent Advances in Natural Language Processing*. 2013.
42. Ritter A. Extracting knowledge from Twitter and the Web. PhD thesis. 2012.
43. Pinto, Oliveira A. Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text. *OASlcs OpenAccess Ser Informatics*. 2016; 16.
44. Kudoh T, Matsumoto Y. Use of support vector learning for chunk identification. 2000. p. 142.
45. Li X, Roth D. Exploring evidence for shallow parsing. 2001. pp. 1–7.
46. Nararatwong R, Kertkeidkachorn N, Cooharajanane N, Okada H. Improving Thai word and sentence segmentation using linguistic knowledge. *IEICE Trans Inf Syst*. 2018; E101D: 3218–3225. <https://doi.org/10.1587/transinf.2018EDP7016>
47. Rei M. Semi-supervised Multitask Learning for Sequence Labeling [Internet]. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.
48. Ponvert EF. Unsupervised Partial Parsing. 2011.
49. Ponvert E, Baldridge J, Erk K. Simple unsupervised identification of low-level constituents. *Proceedings—2010 IEEE 4th International Conference on Semantic Computing, ICSC 2010*. 2010. pp. 24–31.
50. Solan Z, Horn D, Ruppin E, Edelman S. Unsupervised learning of natural languages. *Proc Natl Acad Sci*. 2005; 102: 11629–11634. <https://doi.org/10.1073/pnas.0409746102>
51. Indig B, Endrédy I. Gut, besser, chunker—selecting the best models for text chunking with voting. *International Conference on Intelligent Text Processing and Computational Linguistics. Springer*; 2016. pp. 409–423.
52. Goldsmith J. Unsupervised learning of the morphology of a natural language. *Comput Linguist*. 2001; 27: 153–198. <https://doi.org/10.1162/089120101750300490>
53. Febres G, Jaffe K. Music viewed by its Entropy content: A novel window for comparative analysis. *PLoS One*. 2017; <https://doi.org/10.1371/journal.pone.0185757> PMID: 29040288
54. Febres G, Jaffé K. Calculating entropy at different scales among diverse communication systems. *Complexity*. 2016; 21.

55. Tshitoyan V, Dagdelen J, Weston L, Dunn A, Rong Z, Kononova O, et al. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*. 2019 Jul; 571(7763):95–8 <https://doi.org/10.1038/s41586-019-1335-8> PMID: 31270483
56. Febres G, Jaffe K. A Fundamental Scale of Descriptions for Analyzing Information Content of Communication Systems. *Entropy*. 2015; 17: 1606–1633. <https://doi.org/10.3390/e17041606>
57. Seginer Y. Fast Unsupervised Incremental Parsing. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. 2007. pp. 384–391.
58. Ponvert E, Baldridge J, Erk K. Simple Unsupervised Grammar Induction from Raw Text with Cascaded Finite State Models. *Comput Linguist*. 2011; 1077–1086.
59. Klein D, Manning CD. A generative constituent-context model for improved grammar induction. 2001. p. 128.
60. Headden WP, Johnson M, McClosky D. Improving unsupervised dependency parsing with richer contexts and smoothing. 2009. p. 101.
61. Cohen SB, Smith NA. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. 2009. p. 74.
62. Abney SP. *Parsing By Chunks*. 2011. pp. 257–278.
63. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed Representations of Words and Phrases and their Compositionality. 2013.
64. Jang B, Kim I, Kim JW. Word2vec convolutional neural networks for classification of news articles and tweets. *PLoS One*. 2019; 14: e0220976. <https://doi.org/10.1371/journal.pone.0220976>
65. Lightfoot D. *The development of language: Acquisition, change, and evolution* [Internet]. Oxford: Blackwell; 1999.
66. Cocho G, Flores J, Gershenson C, Pineda C, Sánchez S. Rank diversity of languages: Generic behavior in computational linguistics. *PLoS One*. 2015; 10: e0121898. <https://doi.org/10.1371/journal.pone.0121898>
67. Ritter A, Clark S, Mausam, O E, Etzioni O. Named Entity Recognition in Tweets: An Experimental Study. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*; 2011. pp. 1524–1534.
68. Chinchor N, Sundheim B. MUC-5 EVALUATION METRIC S. 1992.
69. Sasaki Y, Fellow R. The truth of the F-measure. 2007.
70. Marcus M, Santorini B, Marcinkiewicz M. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput Linguist*. 1993; 19: 313.
71. Pitler E, Bergsma S, Lin D, Church K. Using web-scale N-grams to improve base NP parsing performance. *Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics*; 2010. pp. 886–894.