Automatic Detection and Visualization of Information Structure in English

John Blake
University of Aizu
Aizuwakamatsu, Japan
jblake@u-aizu.ac.jp

Evgeny Pyshkin
University of Aizu
Aizuwakamatsu, Japan
pyshe@u-aizu.ac.jp

Šimon Pavlík
Romanesco
Prague, Czech Republic
simon.pavlik@romanesco.hk

ABSTRACT

This paper describes the design and development of an online tool that identifies and visualizes information structure in user-submitted texts written in English. Non-native users of English find it difficult to distinguish between structures that are marked and unmarked. Markedness is evaluated based on acceptability and frequency of a sequence of word tokens. Marked sentences stand out as being unnatural to native speakers, but few native speakers can explain why. Information structure can, however, frequently explain markedness. The tool detects the three principles of information structure: information focus, information flow and end weight. Information focus explains the sequence of elements within sentences. Information flow explains the sequence of elements within paragraphs. End weight explains the relative position of phrases and clauses within a sentence. Through exposure to these principles in context, this tool aims to help writers of English understand which structural language features may be judged as marked.

CCS CONCEPTS

• **Applied computing** → **Document searching**; **E-learning**.

KEYWORDS

information structure, online learning platform

ACM Reference Format:
John Blake, Evgeny Pyshkin, and Šimon Pavlík. 2022. Automatic Detection and Visualization of Information Structure in English. In *2022 6th International Conference on Natural Language Processing and Information Retrieval (NLPIR 2022), December 16–18, 2022, Bangkok, Thailand*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3582768.3582784

1 INTRODUCTION

1.1 Background

Non-native writers of English may be able to draft sentences that are grammatically accurate and lexically appropriate, yet the syntactic and structural choices selected are dissimilar to those of native speakers.

NLPIR 2022, December 16–18, 2022, Bangkok, Thailand

ACM ISBN 978-1-4503-9762-9/22/12…$15.00
https://doi.org/10.1145/3582768.3582784

1.2 Markedness

Substantial exposure to a language enables users to evaluate which syntactic and structural choices are more frequently used and so considered unmarked (or natural); and those which are infrequently or never used and so considered marked (or unnatural). The ability to write in an unmarked manner is an indicator of high-level proficiency.

The first example of marked usage was taken from a corpus of draft research articles written by postgraduate students with English as an additional language. The revised version shows how the same sentence can be written to create a more acceptable unmarked usage [4].

(1) The concepts in the examples of computational functions are two.

(2) There are two concepts in the examples of computational functions.

What is challenging, however, is to explain why the revised version is unmarked and the initial version is marked. The central difference is frequency. The more frequently particular sequences of words are used, the less marked the sequence is. However, it is not possible for learners without extensive exposure to English to assess whether particular permutations of word tokens are frequently used.

1.3 Governing Principles

Information structure [2], which is a set of organisational principles, however, may be used to explain markedness; and so one way to help advanced learners of English to draft less marked texts is to help them understand these principles. In the same way as grammars are created to explain the sequencing of words into phrases, clauses and sentences; information structure helps explain the choices that are made in sequencing information. In the simplest case, there are two pieces of information *A* and *B*. This provides learners with two possible sequences, namely *AB* or *BA*. The principle of information focus governs this choice.

Japanese users of English as an additional language who need to write scientific articles in English face an onerous challenge. This challenge is exacerbated when the learners have little exposure to scientific articles written in English and seldom function in English. Without significant exposure to English, learners are unlikely to be able to evaluate whether grammatically accurate sentences are marked or unmarked, and so increasing their ability to understand the principles of information structure provides them with a method to evaluate markedness and be better positioned to draft unmarked texts [4].

1.4 Contribution

To address the difficulty of markedness, the Language Feature Detector was created. This pedagogic online tool harnesses the power of the web [6] to help second language writers of English understand the impact of the three principles of information structure on markedness. Given the different theoretical perspectives, annotation of information structure is non-trivial and is a topic of current debate [5]. However, rather than advocating a particular theoretical standpoint, we adopt a practical pedagogic approach, putting the learner first. Although some computational linguists have addressed information structure, the components of their structural schemes tend to focus on move structure [8] rather than the principles of information structure.

1.5 Overview

Section 2 starts with a general overview of information structure, and provides a brief linguistic introduction to the three principles of information focus, information flow and end weight. Section 3 describes the priority given to learner needs and how aspects of the three principles of information structure are operationalized. Section 4 provides the technical details related to the software. Section 5 gives a preliminary evaluation of user response based on a small-scale pilot study, summarizes the contribution, and suggests areas for future work.

2 INFORMATION STRUCTURE

Having learned the typical grammatical structures, learners tend to view the sequencing of words as an open choice providing no grammatical constraints are broken. However, collocation and collostruction impact word choice [13]. Word collocations vary from inseparable to loosely connected. This idea of a restricted rather than an open choice was first noted by Sinclair in his principle of idiom [11]. Information structure is the general name for three closely connected principles: information focus, information flow and end weight. The ways in which information structure is used differs with languages [12] and is therefore a research area in translation studies [10].

2.1 Information Focus

In English the focus is usually placed on information at the end. Thus, we tend to provide new information in this position. This end-focus form is unmarked. Initial focus is used to bring the readers attention to the start of the sentence. There are two common ways of changing the focus from end to initial. The first is by bring a clausal element from its expected position in the subject-verb-object-complement-adverbial (SVOCA) chain to the front. This is particularly common when a time adverbial, such as *today* or *yesterday* is fronted. Another way is to use inversion which involves the fronting of negative adverbials and inverting the grammatical subject and operator. This is most commonly used as a rhetorical device.

2.2 Information Flow

Within a sentence, there is a tendency for information to flow from what is understood or known to what is new. This aligns closely with the *default* setting of information focus. However, information flow also describes thematic development within a paragraph. The grammatical theme of a sentence may or may not be derived from the theme or rheme of the previous sentence, resulting in constant, linear or ruptured themes [7]. Constant themes start with the same theme as the previous sentence, linear themes occur when the rheme becomes the theme while ruptured themes occur when an unrelated theme is introduced. Learners of English have seldom analyzed how and where information is described, and so making the thematic structure explicit through labelling can help show where writers may be missing the opportunity to develop an argument. Using passive voice when describing processes is one example of using a constant theme to make a complex process easy to understand because the theme of subsequent sentences is the same. Agent phrases in passive voice constructions contain new information in approximately 90% of cases, thus enabling the default end-focus [2].

2.3 End Weight

Complicated elements are usually placed at the end of a phrase, clause or sentence [14]. Typically, these contain new information and are the focus of the reader. Thus, when sequencing prepositional phrases in a sentence, the longer more complex phrase will occur at the end when the end weight principle is followed. In academic and scientific writing, long complex noun phrases are often used as grammatical subjects, resulting in front-heavy sentences that are difficult to follow for lay readers. Delaying complex elements reduces the burden on working memory and results in a more reader-friendly text [1]. Some common ways to realise end weight include extraposition by use of a cataphoric reference and postponement by delaying the tail of a noun phrase resulting in a discontinuous phrase.

3 DESIGN

As this is a pedagogic tool, the needs of the learners were prioritized. Based on focus group discussions, we decided to concentrate on the aspects of information structure that learners could most easily understand and implement in their own writing. In general, learners expressed a desire to see how texts that they draft conform to the principles of information structure. The concepts we selected as worthy of inclusion in the first version of the Language Feature Detector were:

(1) **Information Focus**: Given / New; New / Given; Fronted adverbial (marked); Fronted adverbial(unmarked)
(2) **Information Flow**: Constant theme; Linear theme; Ruptured theme
(3) **End weight**: End Weight (Sentence); Front heavy (Sentence); Front heavy (Adverbial)

The front-end interface to our tool was implemented as a web app. The user interface was designed to be intuitive and minimal. The submission system harnesses a submission form below which the output is displayed. Fig. 1 shows the submission form and the end weight statistics while Fig. 2 shows the labels associated with each sentence submitted.

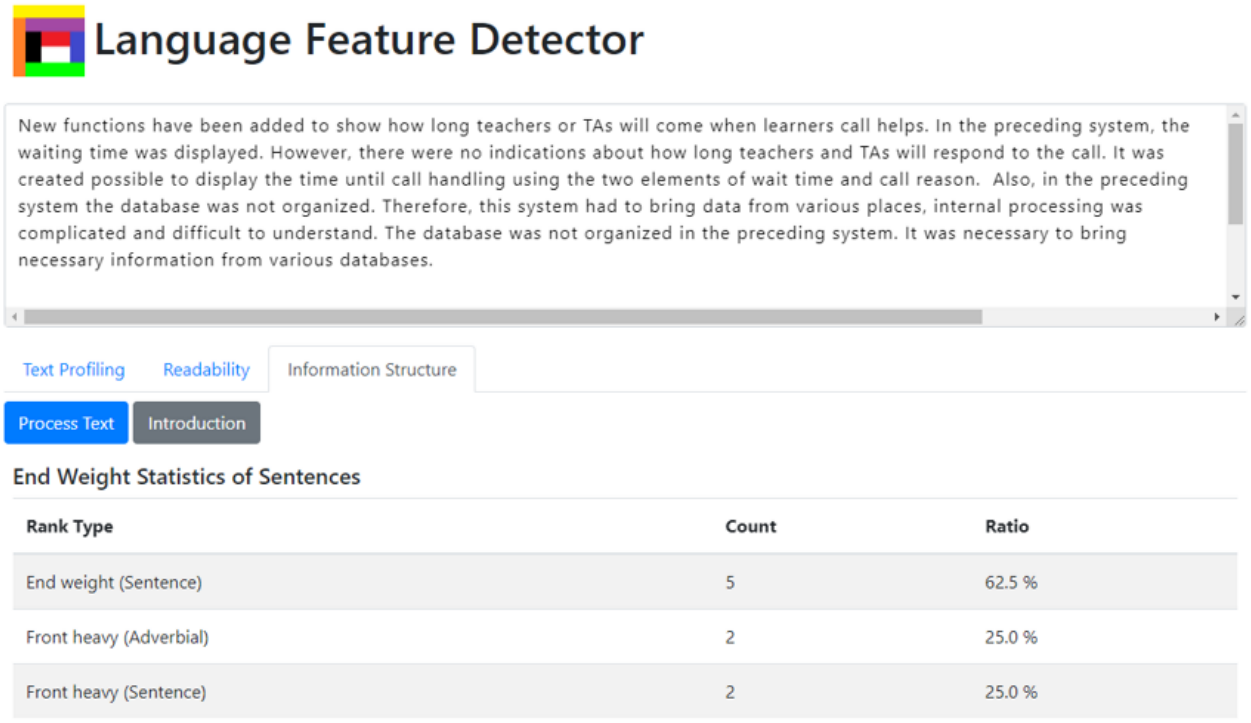

Figure 1: Screenshot of the submission field and end weight statistics

No.	Sentence Text	Information Structure
1	New functions have been added to show how long teachers or TAs will come when learners call helps.	Given/New, End weight (Sentence), Front heavy (Adverbial)
2	In the preceding system, the waiting time was displayed.	Passivized Ruptured Theme, New/Given, Fronted adverbial (unmarked), Front heavy (Sentence)
3	However, there were no indications about how long teachers and TAs will respond to the call.	Ruptured Theme, New/Given, Fronted adverbial (unmarked), End weight (Sentence), Front heavy (Adverbial)
4	It was created possible to display the time until call handling using the two elements of wait time and call reason.	Passivized Ruptured Theme, New/Given, End weight (Sentence)
5	Also, in the preceding system the database was not organized.	Passivized Ruptured Theme, New/Given, Fronted adverbial (unmarked), Front heavy (Sentence)
6	Therefore, this system had to bring data from various places, internal processing was complicated and difficult to understand.	Ruptured Theme, Given/New, Fronted adverbial (unmarked), End weight (Sentence)
7	The database was not organized in the preceding system.	Passivized Ruptured Theme, New/Given
8	It was necessary to bring necessary information from various databases.	Constant Theme, Given/New, End weight (Sentence)

Figure 2: Screenshot of information structure labels for each sentence

4 DEVELOPMENT

The whole information structure module relies on dependency trees of sentences parsed by spaCy [9]. Although we rely on existing libraries for parsing and tokenization, we performed numerous experiments to measure frequencies and fine-tune the bespoke pattern-matching algorithms. The input text is first parsed into simple or complex sentences using Punkt sentence tokenizer by NLTK [3], which performs Unsupervised Multilingual Sentence Boundary Detection. The default method used by spaCy for sentence tokenization results in sub-sentence structures or clauses, each having a single root represented by the main verb. This is utilized to further analyse the structure of each sentence. The front-end is provided with sentence spans and all identified applicable labels. For end-weight we also include frequencies of each of the labels excluding the extraposition label.

4.1 Information Focus

4.1.1 Given/New. The given/new detection depends on the token dependencies provided by spaCy. We only take the first main clause of a complex sentence into consideration. First, we split the sentence into front and rear parts by the main verb (root) of the sentence. Then we search for the pre-defined rear and front side dependency patterns listed below.

- **Front**: `nsubjpass, auxpass; npadvmod, punct, expl`
- **Rear**: `poss, attr; det, attr; nummod, amod, attr, prep`

These particular patterns are located in a sequence of consecutive tokens of a sentence and cannot contain different tokens in between. This implementation of the tool, however, allows us to easily add new patterns that can have an arbitrary number of words interrupting the patterns.

4.1.2 Fronting. Using the same principle as in the Given/New detector, we pre-define a set of dependency patterns to be used as part of a fronting detector. The following are the currently defined patterns for fronted adverbials. Three dots (...) represents an arbitrary number of tokens.

- `advcl, ..., punct, ..., root`
- `npadvmod, ..., punct, ..., root`
- `advmod, ..., punct, ..., root`
- `npadvmod, nsubj, ..., root`
- `prep, det, pobj, punct, ..., root`
- `prep, det, pobj, prep, pobj, punct, ..., root`
- `prep, det, pobj, prep, det, pobj, punct, ..., root`
- `prep, pobj, punct, ..., root`
- `prep, compound, compound, pobj, punct, ..., root`
- `prep, nummod, pobj, prep, pobj, punct, ..., root`
- `pobj, punct, ..., nsubjpass, ..., root`
- `, punct, ..., nsubj, ..., root`
- `prep, nummod, pobj, punct, ..., root`
- `prep, amod, pobj, punct, ..., root`

We also defined anti-patterns that are checked before the regular patterns to avoid their potential subsets to be falsely detected. Currently checked anti-patterns are given below:

- `nsubj, punct, advmod, punct, root`

4.1.3 Markedness. To evaluate the markedness of the detected fronted adverbial, we estimated the probability of each pattern occurring in a text. In order to obtain the probabilities, we measured the number of sentences with given pattern occurrences in the Brown Corpus. Based on the observed frequencies, the markedness threshold was set to 1% probability of occurrence of a sentence with a fronted adverbial pattern in the text.

4.2 Information Flow

We capture the information flow between each consecutive pair of sentences, where the second sentence in each pair is labelled. In the current implementation, we only collect the first main clause of each complex sentence as extracted by spaCy. We rely on several dependency tree based rules defined for the extraction of subject words and non-subject (object) words (S/O) from a sentence clause.

4.2.1 S/O Extraction. The first step consists of selecting the correct position for each S/O within the dependency tree structure of a sentence. The S/O must be a direct descendant of the root (i.e. main verb) although auxiliary passive form or an agent type dependency may occur between. Both subjects and objects are divided into two types: simple and clausal. A Simple type consists of a single word only while Clausal types refer to whole clauses requiring further processing to extract individual S/O words. The dependency types used for the simple objects and subjects are:

- objects: `dobj, pobj; clausal: ccomp`
- subjects: `nsubj, nsubjpass; clausal: csubj, csubjpass`

In the next step, the simple-dependency type tokens are extracted directly, while for clausal ones we extract all its children that are `nsubj` or its `conjs`.

4.2.2 Pronoun Conversion of S/O in the First Sentence. Considering the situation where S/O in the first sentence is referred to by an anaphoric pronoun in the subsequent sentence, we need to be able to match such combinations between sentences. Depending on the part of speech of the words in the first sentence we apply different techniques to map these S/O to their corresponding pronouns to allow for comparison with potential second sentence pronouns. If the entity type of noun is `Person`, we apply a trained gender classifier to determine the gender of his/her name. We use the Naive Bayes classifier implementation by NLTK with name suffixes as input features (last 1 and 2 characters of the name). This is trained on the NLTK names corpus resulting in testing accuracy of approximately 80%. For other nouns we check for noun gender based on a list of typical masculine and feminine nouns such as *father/mother* or *sultan/sultana*. We determine whether a noun is plural based on the comparison of its lemmatized version with itself using the Wordnet lemmatizer within NLTK. After determining gender and the status of singular or plural, a simple mapping is used to transform the noun into a pronoun.

4.2.3 S/O Matching Rules between Two Sentences. The most simple rule is to check for an exact S/O match between each sentence pair. We also check for noun synonyms by searching Wordnet Synsets using the lemmatized form of the words. The second rule is to match the S/O from the first sentence converted to pronouns beforehand with the original s/o from the second sentence. We check for all possibilities including the cases where the first sentence contains

multiple words and the second sentence has one plural pronoun, i.e. *they* or *them*. We also include mixed cases, where the first sentence contains multiple words combined with *I, me* or *you* and the second sentence contains *we*, *us* or *you*.

4.2.4 Passivized Theme. The passivized theme detector checks each sentence for the presence of `auxpass` dependency and adjusts the label of each classified theme above accordingly.

4.3 End Weight

Once again we rely on the token dependencies from spaCy as well as the dependency tree structure to identify Sentential, Clausal and Adverbial Ranks of a sentence. We also determine the counts and ratios of such sentences within the whole input text.

4.3.1 Sentence Rank. By counting the number of tokens before and after the first main verb (root) in a sentence, we compute its rank. The sentence has *End weight* if the difference between the number of tokens after the root is greater than the number of tokens before the root plus three. If the number of tokens after the root is smaller than before the root, then the sentence is considered *Front heavy*.

4.3.2 Clause Rank. Each clause of a more complex sentence is extracted using spaCy's default pipeline that utilizes dependency trees to break the sentence into clauses each having one main verb (root). Each main clause can further contain `advcl` (Adverbial Clause), `relcl` (Adjective Clause) or `ccomp` (Noun Clause). Apart from a clause located before or after the main clause, we also consider cases where clauses are embedded within a main clause. We count all tokens that belong either to the main clause or the three mentioned clauses. To count their tokens, we consider the main verb of the clause including all its descendants in the dependency tree.

4.3.3 Ascending/Descending. Having collected the lengths of all clauses in a sentence, we check whether the lengths are ascending or descending. They are accepted if the difference between consecutive sizes is less than three tokens. For greater differences, the sizes also need to stay constant or increase/decrease. The monotonicity condition cannot be broken. We label the sentence as having *End Weight (clause)* if the token counts are ascending. We label the sentence as being *Front Heavy (clause)* if the token counts are descending.

4.3.4 Adverbial Rank. The Adverbial clause counts calculation is similar to that of the above-described Clause Rank method. We check for the counts of all tokens that are of the following dependencies or their children:

`advcl, npadvmod, prep, advmod, prt`

We skip the adverbial tokens of an adverbial that is embedded within another adverbial clause. Thereby, we make sure that a larger adverbial clause takes precedence over its individual adverbial tokens. The method and labelling of sentences for adverbials is the same as the ascending/descending approach for clauses above.

4.3.5 Extraposition. Considering again only the first main-clause of a sentence, we label the sentence as having an Empty subject if the following conditions are met. The first word is *it*, the second word is the main verb (root) and in the remainder of the main clause we identify a pattern with dependencies

- `acomp, . . . mark, . . . ccomp`

5 EVALUATION

Positive feedback was received in a small-scale pilot test with three advanced users of English. None had considered information structure before and were particularly interested in the labels related to thematic development. None knew the terms end weight or front-heavy, but found the concepts easy to follow when analyzing the output of the Language Feature Detector.

This pedagogic implementation of the Language Feature Detector aims at providing non-native users of English with information on how texts are organized so that they can make informed choices about how to improve their texts. This version includes the easier-to-understand and easier-to-implement aspects of information structure. Given that no other researchers have attempted to automatically detect information structure, and no similar systems were discovered in an extensive search of the literature, despite its limitations, this system breaks new ground.

Before full release of the online system, a comprehensive analysis of the accuracy and precision of the tool needs to be conducted. With simple texts, the system performs well and serves as a good proof-of-concept implementation; but as the complexity of the language increases, the accuracy of the system falls. This appears mainly due to errors introduced by components in the pipeline, such as part-of-speech tagging and entity recognition.

REFERENCES

[1] Alan Baddeley. 2012. Working memory: Theories, models, and controversies. *Annual Review of Psychology* 63 (2012), 1–29.

[2] Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, Edward Finegan, and Randolph Quirk. 1999. *Longman Grammar of Spoken and Written English.* Longman, London.

[3] Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions.* ACL, Sydney, Australia, 69–72.

[4] John Blake. 2015. Incorporating information structure in the EAP curriculum. In *2nd International Symposium on Innovative Teaching and Research in ESP. Tokyo: University of Electro-Communications.*

[5] Hannah Booth. 2022. Desiderata for the Annotation of Information Structure in Complex Sentences. In *Proceedings of the 16th Linguistic Annotation Workshop (LAW-XVI) within LREC 2022.* 31–43.

[6] Ching-Fen Chang and Chih-Hua Kuo. 2011. A corpus-based approach to online materials development for writing research articles. *English for Specific Purposes* 30, 3 (2011), 222–234.

[7] Lise Fontaine. 2013. *Analysing English grammar: A systemic functional introduction.* Cambridge University Press, Cambridge.

[8] Yufan Guo, Anna Korhonen, Maria Liakata, Ilona Silins, Lin Sun, and Ulla Stenius. 2010. Identifying the information structure of scientific abstracts: an investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing.* 99–107.

[9] Matthew Honnibal. 2012. SpaCy [software]. https://spacy.io/

[10] Ángel Luis Jiménez Fernández. 2019. Information-structure strategies in English/Spanish translation. *Journal of English Studies* 18 (2019), 83–107.

[11] John Sinclair. 1991. *Corpus, Concordance, Collocation.* Oxford University Press, Oxford.

[12] Sanghoun Song. 2014. *A grammar library for information structure.* Ph. D. Dissertation. University of Washington.

[13] Anatol Stefanowitsch and Stefan Th Gries. 2003. Collostructions: Investigating the interaction of words and constructions. *International Journal of Corpus Linguistics* 8, 2 (2003), 209–243.

[14] George Yule. 1998. *Explaining English Grammar: A Guide to Explaining Grammar for Teachers of English as a Second Or Foreign Language.* Oxford University Press.