Castledown

OPEN ACCESS

Australian Journal of Applied Linguistics
ISSN 2209-0959
https://www.castledown.com/journals/ajal/

Australian Journal of Applied Linguistics, 4 (3), 119–131 (2021)
https://doi.org/10.29140/ajal.v4n3.538

Automated identification of discourse markers using the NLP approach: The case of *okay*

ABDULAZIZ SANOSI [a]

MOHAMED ABDALLA [b]

[a] *Prince Sattam bin Abdulaziz University, SAUDI ARABIA*
a.assanosi@psau.edu.sa

[b] *Prince Sattam bin Abdulaziz University, SAUDI ARABIA*
m.abdalla@psau.edu.sa

Abstract

This study aimed to examine the potentials of the NLP approach in detecting discourse markers (DMs), namely okay, in transcribed spoken data. One hundred thirty-eight concordance lines were presented to human referees to judge the functions of okay in them as a DM or Non-DM. After that, the researchers used a Python script written according to the POS tagging scheme of the NLTK library to set rules for identifying cases where okay is used as non-DM. The output of the script was compared to the reference human-annotated data. The results showed that the script could accurately identify the function of okay as DM or non-DM in 92% of the cases. The inaccuracy of detecting the rest was found to be caused by a lack of proper and detailed punctuations. The main implications of the results are that new NLP approaches can detect DMS; however, proper punctuation is required to enable the proper identification of DMs. In accordance with the findings, the researcher recommended adopting the approach after conducting further comprehensive studies.

Keywords: discourse markers, nlp, machine learning, corpus linguistics, discourse analysis

Introduction

In linguistics, discourse is "an utterance whose magnitude is bigger than the sentence" (Jabeen, Rai, & Arif, 2011, p. 69). It usually occurs due to "interaction between two or more participants who start talking to achieve a communicative goal" (Guo, 2015, p. 69). As such an interaction entails that discourse should be coherent and logically presented, language users tend to employ various

Data Availability Statement: *All relevant data are within this paper.*

linguistic elements to maintain the coherence and relevance of their speech to make this interaction more effective. These elements have additional functions such as filling pauses, emphasising specific points, eliciting feedback, and signalling new topics. As the case with their various functions, many terms have been suggested referring to such elements, the most common of which is Discourse Markers (DMs). Researchers investigate and analyse DM using different approaches to explore their function, structure, and distribution in the discourse.

New methods of linguistic analysis have prospered recently, where the vast advances in technology have made using the computer for linguistic analysis a preferable trend. Corpus-based studies and computational linguistics are two disciplines that can stand as a representation of this breakthrough. It is now possible to analyse corpora of millions of words to investigate patterns of language use, frequency of occurrence, words collocation and colligation. When it comes to DMs, however, contextual difficulties that may harden the identification of DMs in the given discourse arise because DMs are extremely context-dependent (Furkó, 2020). For this reason, determining DMs within corpora entails manual analysis, which may violate the core concept behind using a corpus-based approach, i.e., automatic analysis of vast bodies of text.

The current study investigates such a problem and proposes dealing with it by adopting an approach that has emerged by the same advance in technology and tools mentioned above. Natural Language Processing (NLP) is a modern approach whereby linguists can analyse different linguistic structures. The concept of NLP is based on enabling the computer to identify specific structures and linguistics phenomena, taking existing modules as references. These modules or libraries are pre-designed using a programming language such as Python. The researchers suggest that designing a set of python codes to identify DMs within corpora will solve the impossibility of recognising DM automatically. These codes will depend on a set of contextual parameters that will be compiled using Python. After applying the codes to the dataset, the results will be compared against a set of reference results generated by human identification of those markers. It is hypothesised that the results will be comparable.

Background of the Study

Discourse Markers (DMs)

Despite the abundance of studies investigating DMs in recent years, there is no consensus either on one term that refers to them or on their definition and function (Alonso, Castellón, & Padró, (2002). Many rubrics refer to those elements that facilitate the addressees' interpretation of the discourse according to the surrounding context. Among these terms is Discourse Particles, which (Schourup, 1983) used to refer to what had previously been known as fillers, hesitations, and interjections (Fraser, 1996). Another term suggested by Halliday and Hassan (1985) as Sentence Connectives implying that their role is to make the building blocks of a text cohere into one solid logic discourse. Fraser (1996) referred to such elements as Pragmatic Markers assigning pragmatic functions for them, while Aijmer (2002) generalised the term into Discourse particles. However, the term DMs is the most common among these terms (Brinton, 1996) and is used by many researchers.

This discord on a standard term for DMs stems from, and is reflected on, the definition of DMs. However, most definitions of them agree on following a functional approach that focuses on what they do rather than how they are structured. They are perceived as "sequentially dependent elements which bracket units of talk" (Schiffrin, 1987, p. 31), a definition that embraces a large group of linguistic units. (Fraser, 1996) considered DMs a fundamental part of a sentence that, although distinct from the meaning content of the sentence, signal the speaker's potential intention while

(Aijmer, 2002) considered them elements that facilitate the addressees' interpretation of the text according to the surrounding context. Other functions proposed by researchers to be achieved by DMs include expressing "the relation or relevance of an utterance to the preceding utterance or the context" (Brinton, 1996, p. 30), signalling a change in discourse development. (Jabeen, Rai, & Arif, 2011), Furthermore, "marking the speaker's attitudes to the proposition being expressed as well as for facilitating processes of pragmatic inferences" (Furkó, 2020, p. 1). The ultimate aim of DMs, it can be argued, is to "organise and 'manage' quite extended stretches of discourse" (McCarthy, 1991).

Corpus Linguistics

A simple definition to describe Corpus Linguistics is provided by McEnery and Wilson (2001) as "the study of language based on examples from real-life use" (p. 1). It does so by utilising a corpus which is "a large principled collection of naturally occurring examples of language stored electronically" (Bennett, 2010, p. 1). Typically corpora are sampled to "represent a certain language, language variety, or other linguistic domain" (Kübler & Zinsmeister, 2015). From this definition, it can be grasped that this relatively modern approach in linguistics differs from the traditional protocols of analysing language. Descriptive methods of linguistic phenomena relied on inauthentic and maybe even imaginary examples that may never exist in real-life use of language and hence did not represent natural language. Moreover, the large size of corpora could represent language more than a limited set of data can do. Furthermore, the utilisation of computer and related technology can present a more accurate and comprehensive analysis of data.

The word text is now used to describe written and spoken discourse (McEnery & Wilson, 1996) as corpus linguistic studies extended later to include spoken registers. Before, written English was the primary source of information on English structure and other aspects (Sattvic, 2007). Accordingly, corpus linguistics dealt with linguistic analysis at the sentence level and was limited to syntactic analysis that focused on the internal structure of these smaller units. However, because "the new emphasis on studying language-in-use, and the implications for the description of pragmatic functions, have affected how we conceptualise the relationship between form and function in language." (Adolphs, 2008), it is noted that current corpus linguistics studies are generally "considered to be a type of discourse analysis because they describe the use of linguistic forms in context. For example, words are described in terms of their typical collocates" (Biber, Connor, & Upton, 2007, p. 2).

Investigating spoken discourse, however, needs elaboration of supra-sentential features that take account of the discourse as a whole and are not limited to smaller units. The main problem is that "Pragmatic concepts are often harder to specify than syntactic or semantic ones since many of them are not fixed, hard-coded rules, but are rather guidelines of how to convey and interpret language" (Kübler& Zinsmeister, 2015, p. 117). The use of DMs is an excellent example of how these pragmatic features are perceived and analysed. Thus, while it seems convincing to use the corpus linguistics method to determine the frequency and dispersion of DMs within a specific text, the method may overcalculate the instances of DMs simply because it only considers the use of DMs at the sentence level with no consideration to pragmatic features that determine whether the use of the specific word or a phrase was a DM or non-DM. The solution to this dilemma, the researchers believe, lies in a different, though not separable, method of linguistic analysis, which is Natural Language Processing (NLP).

Natural Language Processing

Natural Language Processing (NLP) is a method that refers to the "computational techniques that

process spoken and written human language, as language" (Jurafsky & Martin, 2000). This inclusive definition implies a specific (knowledge of the language) that goes beyond counting the characters of words and the frequency of their occurrence. Instead, its aim extends to "make the machine understand and interpret the human language" (Kulkarni & Shivananda, 2019) through converting text data to binary or number format that make them processable with machines. Before that, patterns from natural languages that determine how humans use a specific natural language should be analysed and defined (Kocaleva, Stojanov, Stojanovik, & Zdravev, 2016). These patterns should be taught to a computer. Ultimately, computers will then be able to analyse, recognise and generate human languages and perform different linguistic tasks automatically. Because of such potentials, Hardeniya, Perkins, Chopra, Joshi, and Mathur (2016) compare NLP to teaching language to the child; however, the former is more demanding as teaching linguistics phenomena to humans is natural while teaching computers these tasks is still a tremendous challenge.

Several fields and disciplines form the base of NLP, such as artificial intelligence, data science, theoretical linguistics, computer science, psychology, and logical mathematics (Kibble, 2013); however, it is commonly referred to as a subfield of linguistics termed as computational linguistics. Recently, NLP has been used to achieve broader tasks related to language disorders. Examples of these are studies that assess cognitive impairment, schizophrenia, and Alzheimer's dementia (Corcoran, & Cecchi, 2020, Yeung, et al., 2021). However, it is mainly related to corpus linguistics as the analysis and investigation of human language are executed on language corpora after implementing different NLP techniques on them such as lemmatising and stemming texts, Part of Speech (POS) tagging, parsing, chunking, machine translation, and speech recognition, (Meurers, 2015; Hardeniya, et al., 2016). NLP is used to automate several linguistic tasks such as error detection, error correction, speech recognition, sentiment analysis and text extraction, to name a few. In doing this, it utilises ready-made tools (known as libraries) that facilitate text processing for average users who have no superior knowledge of computer-related sciences. Examples of these libraries are Natural Language Toolkit (NLTK) (Bird, Klein, & Loper, 2009), SpaCy, and Stanford CoreNLP (Kulkarni & Shivananda, 2019). Of these, NLTK is a leading library because it is easy and it includes most NLP tasks (Hardeniya *et al*., 2016).

Previous studies

The literature on discourse markers is rich, and an abundance of studies investigated the functions and use of DMs by learners and other language users. Nevertheless, most of these studies followed either quantitative methods of analysis that investigate the frequency of DMs in specific contexts or qualitative methods that address the multifunctionality of DMs as they are used by the participants or across different genres. As expected, the corpus linguistic method is widely used in such studies since corpus software can provide accurate statistics of frequency and dispersion of DMs in given texts. Although such studies are remarkably inspiring and their methods are primarily robust, there are still some questions regarding the pragmatic properties of DMs that need extra investigation. Recently NLP techniques have been used to detect, identify, and classify DMs in English and other languages. NLP techniques are based on "a variety of computerised semantic tagging (CST) systems, including artificial intelligence-based, knowledge-based, corpus-based and semantic taxonomy-based systems" (Furkó, 2020, p. 15). This makes the problem of investigating the pragmatics function more apparent as these functions are an excellent example of the complexities of human languages that artificial intelligence applications are not capable of grasping. Therefore, more research is required to help in generating a proper model for automatic DM identification. This can be achieved through utilising more linguistic and metalinguistic features that make the process of determining what DM is and what is not a feasible task.

One of the earliest experiments to utilise Machine Learning (ML) to automate the identification of discourse elements was (Litman, 1996) that aimed to classify cue phrases. The study used ML software to classify cue phrases as discourse or sentential, i.e., as DMs or non-DMs. The researcher used data from previous experiments conducted by him and other authors as training data to achieve its aims. The training data included previously classified cue phrases and their features and training examples from the previous experiments. The features he used to classify cue phrases include prosodic features, e.g., length of the utterance, textual feature, e.g., preceding or succeeding punctuation, and lexical features, e.g., the exact token of the word, e.g. (actually, but, because, etc.). The results showed that ML was an effective technique for generating classification models that are more accurate than manual classification and for improving upon previous data. Although the experiment relied on manually created training data, one of the studies predicted that NLP could be used effectively in discourse analysis to provide new linguistic implications. The work on the features of the cue phrases to prepare accurate training data was relatively plentiful. Modern advances in NLP have made it possible to achieve that in a more convenient method.

Attempting to solve the problem of DMs inconsistency and uneven coverage by manual investigation, Alonso, Castell'on, &Padr' (2002) presented the X-TRACTOR, which was a tool for extracting DMs from plain Spanish texts. They designed their system with no assistance from pre-designed NLP tools. Later, they fed it with an unannotated corpus. The tool used a set of language-independent features such as word position in a sentence and its neighboring words. It also utilised a list of possible DMs in Spanish. The researchers then formulate *if/then rules* to determine possible DMs in the plain text. The X-TRACTOR was found effective in ranking Spanish DMs and finding new DMs in the corpus according to several DM identification parameters. However, there was an abundance of human work over the parameters to determine the likelihood of the DMs, and the results were still not agreed upon as there was no complete unanimous determination of some DMs. As NLP aims at reducing human work, the results entail more improvement to their adopted method.

Zufferey and Popescu-Belis (2004) compared the automatic detection of DMs to human recognition. The study revealed that it was a difficult task for human annotators to decide whether the word *like* was used as DM or non-DM in different occurrences in the dialogue transcription. There was a low agreement level between the human judges. However, the annotators' agreement increased when they were assisted with the prosodic cues by listening to the soundtracks of the dialogues. Furthermore, to disambiguate the functions of like, they used automated means depending on collocation, position in utterance and duration-based features. These features were proved efficient in detecting DMs to an acceptable level of precision (70%). However, for more accuracy, the researchers used a POS tagger to disambiguate the use of *like* and *well* as DM or Non-DM. It was proved that this method was unreliable since the POS tagger was unable to disambiguate the occurrences of *like* in dialogue transcriptions. The main shortcoming in this method is that the parameter was made to exclude non-DM cases based on the POS of the word itself. For example, to exclude *like* when it comes as a verb. This method would perform in a limited way, especially with unannotated corpora. The current research considers this point and adopts a POS tag scheme that analyses the whole utterance and hence considers POS and the surrounding words of the concordance lines.

Another study (Petukhova & Bunt, 2009) used a multidimensional model of the interpretation of communicative behaviour in dialogue to describe the multifunctionality of DMs. They utilised an automatic binary classification of DM vs non-DM and an automatic recognition method of the various functions of discourse markers. Via these techniques, they showed that DMs are multifunctional units that can serve multiple communicative functions simultaneously. Although they

adopted corpora to generate these results, they applied an ML technique to automatically recognise the multiple meanings of DMs depending on prosody, word occurrence, and collocations. This study also lacks parameters pertaining to the whole utterance, such as POS tagging, which take the complete utterance into account and encompasses two of the above features (word occurrence and collocations) plus considering the syntactic structure of the whole utterance.

There are other studies aimed at distinguishing DMs from other discourse metadata units. In this regard, Cabarrão *et al*. (2018) analysed DMs in two corpora of European Portuguese. After they had yielded their results showing that the selection of DMs is domain and speaker-dependent, they applied an automatic technique for further analysis of DMs. They used three acoustic-prosodic feature sets and ML to automatically distinguish between discourse markers, disfluencies, and sentence-like units. In structuring their rules for generating DMs, they used paralinguistic features of structural metadata (punctuation marks, disfluencies, inspirations). Their results showed that DMs are more accessible to classify than disfluencies and that the method is considerably accurate in discriminating DMs from other features. Though this model was successful in that it can be used to discriminate DMs from other units that may be comparable to them, however, the model is not able to identify DMs based on their pragmatic and morpho-syntactic behaviour.

The state-of-the-art NLP studies on DM investigation are not limited to only automatic identification and classification. Recent studies extend the task to predict the might-be DMs and suggest possible DMs to occur between pairs of sentences. For instance, in a recent study (Sileo, Van de Cruys, Pradel, & Muller, 2020), the researchers used a trained model they designed to predict plausible DMs between sentence pairs with known semantics relations according to existing classification datasets. The models were designed with many usage examples. Ultimately, they generated a dataset named DiscSense which is available publicly[1] and capable of predicting DMs for different tasks such as entailment, subjectivity analysis, and sentiment analysis.

The studies on automatic identification of DMs have standard features in that they set specific parameters to enable the adopted ML tool to recognise DMs based on these parameters. Mainly, prosodic features, word position, and collocation are used to determine the DMs or the likely cases of their occurrence. Although there was an acceptable level of success of these tools, the applied features encountered some shortcomings that can be summarised in the immense manual work it requires, low level of accuracy, and limited to a specific style, for example, dialogue. To account for this, the current study applies POS tagging for a previously corpus-generated dataset and utilises pre-set parameters written in Python language to identify DMs automatically. The parameters take account of the POS of each word in the utterance and consider the adjacent word to determine DM according to its syntactic and discursive features.

Methods

The current study is a follow-up study in that it used a previously generated dataset of a study conducted by the first author (Sanosi, 2018). Moreover, it is motivated by the difficulties and the inconsistencies that emerged in detecting DMs by relying only on the corpus linguistics method. It is then hypothesised that adopting the NLP approach would be convenient to identify DMs in corpora.

The motivating study

The basic study used the corpus method to investigate non-native English speakers' use of three DMs, namely *ok*, *well*, and *you know*. It utilised two corpora. The first was Sudanese TEDx Spoken English (STSE), representing non-native speakers of English, while the other was a reference corpus

of native English speakers and was named British TEDx Spoken English (BTSE). As the names imply, both corpora were extracted from TEDx talks events presented by the speakers. Some of the talks were transcribed manually while others were transcribed automatically; however, the transcription of all the texts was validated manually. The statistics of the corpora are shown in Table 1 below.

Table 1 *Statistics of the corpora*

Corpus	Texts	Tokens	Talking Time	Types
STSE	67	108621	12.8 hours	8138
BTSE	54	108618	11.6 hours	9311

The result of the study showed variation in the frequency and distribution of DMs across the two corpora. More importantly, the used corpus linguistics software over recognised multiple occurrences of DMs of the three instances as every occurrence of the selected words was marked regardless of contextual or semantic setting. Further manual finetuning was conducted to categorise the functions of the selected words as DMs or non-DMs. Regarding the word *okay*, the results showed that it was used as presented in Table 2 below.

Table 2 *The use of the word okay in the two corpora*

Phrase/word	BTSE			STSE		
	Overall	As non-DM	%	Overall	As Non-DM	%
okay	42	14	33.3	139	21	15.1

Note. Adapted from Sanosi (2018).

The resultant data was sufficient for the scope of the motivating study. Nevertheless, a subsequent research question emerged regarding the ability of a hard-coded program to identify DMs in a POS-tagged corpus following specific parameters that are written in Python. The study was guided by a single research question: How accurately does the NLTK POS tagging script identify DMs in a corpus?

Determining DMs

To achieve the aim of the present study, two datasets were created. Firstly, 139 concordance lines were presented to human referees to judge if the function of the word okay in each line was a DM or non-DM. The concordance lines were extracted from the STSE corpus using AntConc software (Anthony, 2020), and the referees were four university professors specialising in Applied Linguistics (2) Literature (1) and TESOL (1). It was decided that the results upon which at least three referees agreed would be adopted for the analysis. Finally, 1 line was excluded from the analysis because there was a disagreement on the function of okay because of the speaker's mistake.

Secondly, the researchers defined DMs according to 13 contextual parameters to prepare for automatic judging of DMs. These contextual parameters were set according to the POS tagging list, which is a method to label the words in a text with a definite part of a speech according to its definition and context. The researchers used conditional statements to specify cases where the meant

word should be considered a non-DM while the other cases should be considered DMs. For example, the first rule states that If okay is preceded by a base verb, then okay is non-DM. See appendix A for the list of the hard-coded rules.

The third step was to create a Python script using the set of rules. By then, the contextual parameters were transformed into Python codes using NLTK POS tagger (Bird, Klein, & Loper, 2009). The script was supposed to define DMs with the sentences after passing the same 138 lines to it. To yield the results of the study, the researchers compared the human-annotated dataset to the output of the python script and calculated the accuracy of the script in judging the functions of okay in all the lines.

Results

There was an agreement among the human referees that the word *okay* was used as a DM in most of the cases. Figure 1 below presents the results of the human annotation for the function of *Okay* by the human referees.

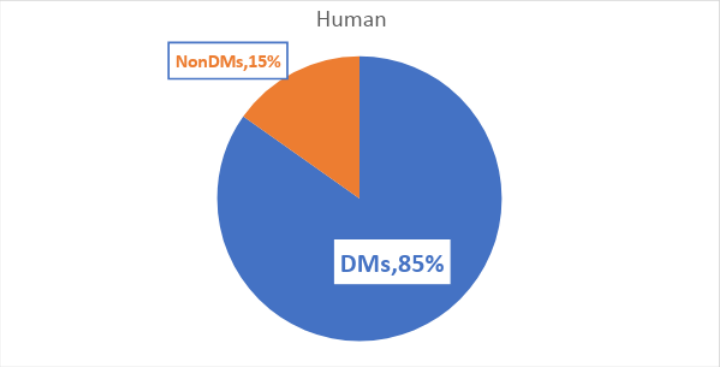

Figure 1 *The functions of **okay** in the human-annotated concordance lines*

The results show that the word *okay* was mostly used as a DM. This result was taken as a reference against which the output of the script would be compared. The Python script, however, generated different results, as Figure 2 displays.

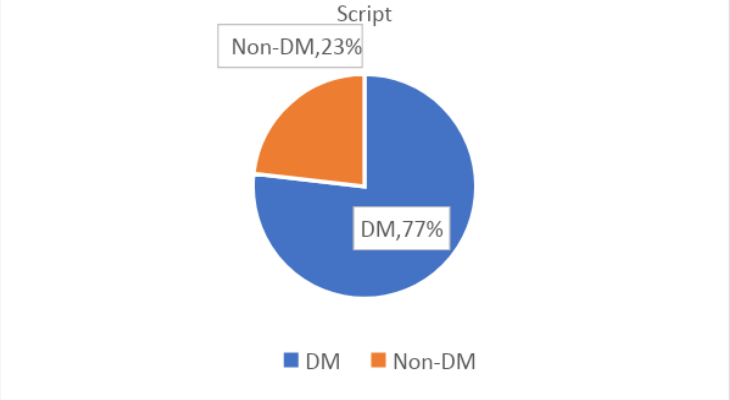

Figure 2 *The functions of okay in the output of the Python script*

Although the results are comparable to the human-annotated lines in that most of the cases of *okay* were DMs, a significant difference (8%) in favor of non-DMs was identified by the script. Taking

into account that the human-annotated lines are the standard, it can be stated that the script overidentified non-DMs in the lines. The calculations show that 11 cases of DMs were considered non-DMs by the script. Table 3 shows the calculations of the results.

Table 3 *The functions of okay in both datasets*

Dataset	DMs	Non-DMs
Human	117	21
Script	106	32
Difference	11	
Accuracy Percent	92.02 %	

The status of each line is considered accurate when the output of the script for the same line is identical to the human judgement in considering the function of *okay* either DM or non-DM. Otherwise, the status of the output is considered inaccurate. As the table shows, the script was successful in identifying the function of the word *okay* in most cases. However, this does not suggest the perfection of the method at this stage, as 8% of inaccuracy is a considerable drawback. Bearing in mind that all the status of inaccuracy were in considering the non-DMs cases, the researchers reviewed the cases where inaccuracy occurred and concluded the following points.

Discussion

The main reason for the shortcomings in identifying the function of *okay* as a DM was the lack of proper punctuation. Most of the texts were punctuated. However, their punctuation was not complete or did not account for all the prosodic and stylistic requirements. The most used marks are commas, periods, and question marks. Moreover, some unpunctuated text files witnessed more cases of inaccuracy. These cases are represented in lack of quotation marks after verbs like *say* or *think* e.g. *"..business through those experiences. Little by little I started to think okay can I make money from this?" (Khalil, 2013).*

"We cannot change your profession from engineering to bars, and I said okay as I come into the university without an invitation." (AbdAlshakur, 2017*).*

In both cases, the word *okay* is considered non-DM by the script following the rules *VB>okay* and *VBD>okay*. Of course, the rule was meant to mark the cases where *okay* is used as an adverb; however, this is not the case in the above examples. Solving this problem can be done by using quotation marks after the words *think* and *said*.

The absence of other common punctuation marks (commas, periods, and question marks) was also spotted and caused further inaccuracy in detecting DMs. For example, line 123 reads:

"Is anybody here Shaigiya? Raise your hand if you are Shaigiya. So OK, we love Shaigiya people." (Abdelrahman, 2012).

The function of *okay* here was considered non-DM since it is preceded by the word *so,* which is considered as an adverb in this context. This problem can be solved by placing a comma after the word *so,* however, it is understandable why the transcriber did not put a comma in such a place. As

punctuation marks are mostly put according to grammatical requirements (to show sentence boundaries, independent clauses, introductory phrases, and so forth), it is unlikely that the transcriber would think of the two words as a DM, that is, a pause-filler. Transcription of modern-media speech, both manual and automatic, considers the exact punctuation requirements, not the prosodic features of the utterance. Although transcription of this type is very helpful for readers and researchers to use such texts, serious problems can arise when the aim is to analyse the whole discourse, including the prosodic features that can change the meaning of some chunks of the speech.

Lack of proper punctuation can also make distinguishing features to identify DMs not reliable. For example, some previous studies on automatic identification of DMs adopt word position in the sentence as (gold standard) to determine if a word is a DM or non-DM. Furkó (2020) states that initial position can serve as an indicator of being a DM in the majority of cases especially in spoken discourse. As periods are used to mark the end of a sentence and the start of a new one, their absence will make it hard to figure out word position accurately.

To sum up, it can be stated that punctuation is the backbone in the process of automatic identification of DMs in texts. This can be justified by recognising that DMs are hard to be judged by only their position in the sentence or their parts of speech of their neighboring words. Consequently, it should be noted that what is meant here is accurate and detailed punctuation that takes account of not only the grammatical structure of sentences. Although this may be a hard-coding style, the researchers believe it will give more efficacy in detecting DMs and solve the problems of both detecting DMs manually or through regular corpus linguistics software.

These findings could be of interest to practitioners and researchers in the field of NLP and corpus linguistics in that they could improve textual annotation to reduce the shortcomings of automatic identification of linguistic units such as DMs. It is crucial for transcribers and corpus annotators, for example, to offer a great deal of attention to metalinguistic features while they transform spoken utterances to annotated texts. Punctuation marks used by them should take account of not only grammatical structure but they should also mark prosodic features and unexpected speakers' behaviors such as repetition and disfluencies. Using a punctuation scheme of this type will make the performance of the automatic identification tool more accurate. As far as the researchers are concerned, it can be stated that no single feature can be applied alone to identify DMs automatically. Therefore, future research suggestions could be to adopt a scheme that combines as many features as possible, i.e., POS tagging, prosodic clues, collocation, and word position. NLTK is proven efficient for POS tagging, and proper punctuation can help to a far extent. Therefore, a research project which applies these criteria is believed to yield better results. The findings of this study are in line with the previous literature in that DMs are hard to detect automatically without firstly annotating (and punctuating) the corpus thoroughly. As the data for the present study is relatively small, the researcher suggests conducting similar research that adopts a larger amount of data that are appropriately annotated. Focusing on the different contexts of spoken discourse will also be more suggestive as the nature of the style and register can affect both the size and type of the used DMs.

The output of successful automatic identification of DMs can be of great value for learners and teachers of the English language, especially for foreign or second language learners. Students nowadays utilise many modern methods to learn English, such as machine translation and speech recognition, to name a few. Proper automatic detection of DMs and similar linguistic units could make English language learning more convenient and valuable. Teachers can use the generated application to teach DMs and punctuation and to evaluate students' academic writing.

Conclusion

This study aimed at investigating the possibility of automatic identification of DMs by using a Python script that utilises NLTK POS tagging. The rationale behind the study was to avoid the shortcomings of standard corpus linguistic software in detecting DMs, that is, over-generalisation. After comparing the script output to the human reference data, it was found that the script could identify most of the occurrences of the word *okay* as done by the human referees. However, there was an 8% of inaccuracy that occurred due to insufficient punctuation. This lack of proper punctuation makes it challenging to determine grammatical and prosodic features of the utterances and hence hinders the accurate automatic identification of DMs. This implies that automatic identification requires more than POS tagging. Transcription of spoken data should be marked with full punctuation that considers disfluencies, repetition, and other phenomena of spoken discourse. The researchers recommend applying these strategies to a larger amount of data and using the output as training data that can generate an ML module to automate the process of detecting DMs in transcribed spoken corpora. Achieving this is believed to generate computer applications that are capable of detecting and evaluating the use of DMs, which would be beneficial for teachers and learners.

References

Abdalshakur, M. (2017, October 24). *Human inspiration (have home of hope)*. TEDx Almogran. Retrieved from https://www.youtube.com/watch?app=desktop&v=kTuPPn_5BoM

Abdelrahman, R. (2012, December 24). Racism in Sudan. TEDx Youth@Khartoum. Retrieved from https://www.youtube.com/watch?v=hcTIswZQQW0

Adolphs, S. (2008). *Corpus and context: Investigating pragmatic functions in spoken discourse.* Amsterdam: John Benjamins.

Alonso, L., Castellón, I., & Padró, L. (2002). X-TRACTOR: A Tool for extracting discourse markers. *Proceedings of LREC Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*.

Anthony, L. (2020). *AntConc* (Version 3.5.9) [Computer Software]. Waseda University. Retrieved from https://www.laurenceanthony.net/software

Bennett, G. R. (2010). *Using corpora in language learning classroom: Corpus linguistics for teachers*. Michigan: University of Michigan.

Biber, D., Connor, U., & Upton, T. (2007). *Discourse on the move: Using corpus analysis to describe discourse structure.* Amsterdam: John Benjamins.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python.* Sebastopol: O'Reilly.

Cabarrão, V., Moniz, H., Batista, F., Ferreira, J., Trancoso, I., & Mata, A.I. (2018). Cross-domain analysis of discourse markers in European Portuguese. *Dialogue & Discourse, 9*(1), 79–106.

Corcoran, C., & Cecchi, G. (2020). Using language processing and speech analysis for the identification of psychosis and other disorders. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 5(8), 770–779. https://doi.org/10.1016/j.bpsc.2020.06.004

Furkó, P. (Ed.) (2020). Preliminary issues: Category membership, methodology, alternative perspectives on discourse markers. In P. Furkó (Ed.), *Discourse markers and beyond: Descriptive and critical perspectives on discourse-pragmatic devices across genres and languages* (pp. 1–35). London: Palgrave Macmillan. https://doi.org/10.1007/978-3-030-37763-2

Guo, F. (2015). A review of discourse markers from the functional perspective. *Journal of Arts and Humanities, 4*(4), 69–75. https://doi.org/10.18533/journal.v4i4.685

Hardeniya, N., Perkins, J., Chopra, D., Joshi, N., & Mathur, I. (2016). *Natural language processing: Python and NLTK*. Birmingham: Packt Publishing.

Jabeen, F., Rai, A., & Arif, S. (2011). A corpus-based study of discourse markers in British and Pakistani speech. *International Journal of Language Studies*, 69–86.

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Hoboken, NJ: Prentice-Hall.

Khalil, Mazin. (2013, December 31). My journey as an entrepreneur. *TEDx University of Khartoum*. Retrieved from https://www.youtube.com/watch?v=ORYAORkBzRw&lc=UggOp39bfGDsyHgCoAEC

Kibble, R. (2013). *Introduction to natural language processing*. London: University of London.

Kocaleva , M., Stojanov, D., Stojanovik, I., &Zdravev, Z. (2016). Pattern recognition and natural language processing: State of the art. *TEM Journal*, 5(2), 236–240.

Kulkarni, A., & Shivananda, A. (2019). *Natural language processing recipes unlocking text data with machine learning and deep learning using Python.* New York: Springer Science.

Kübler, S., & Zinsmeister, H. (2015). *Corpus linguistics and linguistically annotated corpora*. London: Bloomsbury Academic.

Litman, D.J. (1996). Cue phrase classification using machine learning. *Journal of Artificial Intelligence Research*, 5, 53–94.

McEnery, T., & Wilson, A. (1996). *Corpus linguistics: An introduction.* Edinburgh: Edinburgh University Press.

Meurers, D. (2015). Learner corpora and natural language processing. In S. Granger (Ed.), *The Cambridge handbook of learner corpus research* (pp. 537–566). Cambridge: Cambridge University Press.

Petukhova, V., & Bunt, H. (2009). Towards a multidimensional semantics of discourse markers in spoken dialogue. In *Proceedings of the 8th International Conference on Computational Semantics* (pp. 157-168). Tilburg: International Conference on Computational Semantics.

Sanosi, A. B. (2018). Discourse markers in Sudanese spoken English: A corpus-based study. *Journal of Applied Linguistics and Language Research*, 5(6), 74–88.

Schourup, L. C. (1983). *Common discourse particles in English conversation*. Ohio: Ohio State University.

Sileo, D., Van de Cruys, T., Pradel, C., & Muller, P. (2020). DiscSense: Automated semantic analysis of discourse markers. *The 12th Conference on Language Resources and Evaluation* (pp. 991–999). Marseille: European Language Resources Association (ELRA).

Svartvik, J. (2007). Corpus linguistics 25+ years on. In R. Facchinetti (Ed.), *Corpus Linguistics: 25 Years on* (pp. 11–26). New York: Rodopi.

Yeung, A., Laboni, A., Rochon, E., Lavoie, M., Santiago, C., Yancheva, M., Novikova, J., Xu, M., Robin, J., Kaufman, L.D., & Mostafa, F. (2021). Correlating natural language processing and automated speech analysis with clinician assessment to quantify speech-language changes in mild cognitive impairment and Alzheimer's dementia. *Alzheimer's Research & Therapy*, 13(109), 1–10. https://doi.org/10.1186/s13195-021-00848-x

Zufferey, S., & Popescu-Belis, A. (2004). Towards automatic identification of discourse markers. *The 5th SIGdial Workshop on Discourse and Dialogue* at HLT-NAACL (pp. 63–71). Massachusetts Association for Computational Linguistics.

1. DiscSense is available at: https://github.com/synapse-developpement/DiscSense

Appendixes

Appendix A. Suggested Rules for Identifying DMs by the Python Script

#	RULE	EXPLANATION	EXAMPLE
1	VB > okay	If okay is perceded by a base verb then Okay is Non DM.	it is okay/ they look okay
2	DET > okay > NN	If okay is perceded by a determiner And followed by a singular noun then Okay is Non DM.	an okay person
3	DET > okay > NNS	If okay is perceded by a determiner And followed by a plural noun then Okay is Non DM.	the okay chances
4	VBD> okay	If okay is perceded by a past verb then Okay is Non DM.	they were okay / She sounded okay
5	VBZ > okay	If okay is perceded by a 3rd p verb then Okay is Non DM.	it seems okay.
6	PRP$ > okay	If okay is perceded by a possessive adjective then Okay is Non DM.	my okay friends
7	RB > okay	If okay is perceded by an adverb then Okay is Non DM.	it was very okay, more okay
8	VB > okay > NN	If okay is perceded by a base verb And followed by a singular noun then Okay is Non DM.	it is okay man.
9	VBD > okay > NNS	If okay is perceded by a past verb And followed by a singular noun then Okay is Non DM.	they were okay boys.
10	VBZ > okay > NNS	If okay is perceded by a 3rd p verb And followed by a singular noun then Okay is Non DM.	it seems okay brother.
11	PRP > okay ?	If okay is perceded by a pronoun And followed by a question mark then Okay is Non DM.	are you okay?
12	NN > okay?	If okay is perceded by a singular noun And followed by a question mark then Okay is Non DM.	was the school okay?
13	NNS > okay?	If okay is perceded by a plural noun And followed by a question mark then Okay is Non DM.	were the movies okay?

Appendix B. Python Script for Identifying DMs in the Concordance Lines

```
data = pd.read_excel('./Data/allData.xlsx')
```

```
result.clear()
for i, row in data.iterrows():
    classifier(i, row.type, row.sent, row.label)
```

```
def classifier(sentNo, sentType, fullSent, label):
    deconSent = decontracted(fullSent)
    tokenizedSent = nltk.word_tokenize(deconSent.lower())
    sent = nltk.pos_tag(tokenizedSent)
    fullSent = fullSent.strip()
    for i, token in enumerate(sent):
        if i < len(sent)- 1:
            if token[0] == 'okay' or token[0] == 'ok':

                if sent[i - 1][1] == 'DET' and sent[i + 1][1] == 'NN':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'DET' and sent[i + 1][1] == 'NNS':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'VB':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'VBD':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'VBZ':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'PRP$':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'RB':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'VB' and sent[i + 1][1] == 'NNS':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'VBD' and sent[i + 1][1] == 'NNS':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'VBZ' and sent[i + 1][1] == 'NNS':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'PRP' and sent[i + 1][0] == '?':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'NN' and sent[i + 1][0] == '?':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 1][1] == 'NNS' and sent[i + 1][0] == '?':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                elif sent[i - 2][1] == 'PRP' and sent[i - 1][1] == 'VBP':
                    storResult(sentType, sentNo, i, fullSent, label, 0)
                else:
                    storResult(sentType, sentNo, i, fullSent, label, 1)

```

```
def decontracted(phrase):
        # specific
        phrase = re.sub(r"won't", "will not", phrase)
        phrase = re.sub(r"can\'t", "can not", phrase)

        # general
        phrase = re.sub(r"n\'t", " not", phrase)
        phrase = re.sub(r"\'re", " are", phrase)
        phrase = re.sub(r"\'s", " is", phrase)
        phrase = re.sub(r"\'d", " would", phrase)
        phrase = re.sub(r"\'ll", " will", phrase)
        phrase = re.sub(r"\'t", " not", phrase)
        phrase = re.sub(r"\'ve", " have", phrase)
        phrase = re.sub(r"\'m", " am", phrase)
        return phrase
```