frontiers in Digital Humanities

ORIGINAL RESEARCH
published: 17 May 2018
doi: 10.3389/fdigh.2018.00010

Rhetorical Figure Detection: Chiasmus, Epanaphora, Epiphora

Marie Dubremetz and Joakim Nivre*

Department of Linguistics and Philology, Uppsala University, Uppsala, Sweden

Rhetorical figures are valuable linguistic data for literary analysis. In this article, we target the detection of three rhetorical figures that belong to the family of repetitive figures: chiasmus (I **go** where I **please**, and I **please** where I **go**.), epanaphora also called anaphora ("**Poor old** European Commission! **Poor old** European Council.") and epiphora ("This house is **mine**. This car is **mine**. You are **mine.**"). Detecting repetition of words is easy for a computer but detecting only the ones provoking a rhetorical effect is difficult because of many accidental and irrelevant repetitions. For all figures, we train a log-linear classifier on a corpus of political debates. The corpus is only very partially annotated, but we nevertheless obtain good results, with more than 50% precision for all figures. We then apply our models to totally different genres and perform a comparative analysis, by comparing corpora of fiction, science and quotes. Thanks to the automatic detection of rhetorical figures, we discover that chiasmus is more likely to appear in the scientific context whereas epanaphora and epiphora are more common in fiction.

Keywords: rhetorical device, antimetabole, chiasmus, epiphora, epanaphora, repetitive figures, computational stylistics

OPEN ACCESS

Edited by:
Stan Szpakowicz,
University of Ottawa, Canada

Reviewed by:
Vivi Nastase,
Universität Heidelberg, Germany
Julian Brooke,
University of Melbourne, Australia
Anna Kazantseva,
National Research Council Canada (NRC-CNRC), Canada

****Correspondence:***
Marie Dubremetz
marie.dubremetz@lingfil.uu.se

Specialty section:
This article was submitted to Digital Literary Studies, a section of the journal Frontiers in Digital Humanities

Received: *14 August 2017*
Accepted: *30 April 2018*
Published: *17 May 2018*

Citation:
Dubremetz M and Nivre J (2018) Rhetorical Figure Detection: Chiasmus, Epanaphora, Epiphora. Front. Digit. Humanit. 5:10. doi: 10.3389/fdigh.2018.00010

1. INTRODUCTION

Computer science and literature have different cultures (Hammond et al., 2013). Despite that fact, they have something in common: literature and discourse analysis, like the hard sciences, are in need of data. A literature or discourse analyst need data to support their interpretation of a text. These data are not picked randomly: they must be based on well-chosen parts of the text. One of the aspects that may be studied by the analyst are the figures of speech (Pasanek and Sculley, 2008). Figures of speech are known in computational linguistics for the challenge they represent, but only a small subset of them has been recurrently studied in the computational linguistics community, for instance sarcasm and metaphor (Dunn, 2013).

However, there are many more figures of speech besides sarcasm and metaphor. A recently compiled ontology lists more than 70 of them (Kelly et al., 2010). Among them we distinguish a category called the figures of repetition. The figures of repetitions are a family of figures. They can involve repetition of any linguistic element, from sound, as in rhyme, to concept and ideas, as in pleonasm and tautology. In this article we will focus only on figures involving repetition of words: chiasmus, epanaphora, epiphora.

The chiasmus of words, also called antimetabole, is defined as the repetition of a pair of words in reverse order. It is called "chiasmus" after the Greek letter χ because of the cross this letter symbolizes (see **Figure 1**).

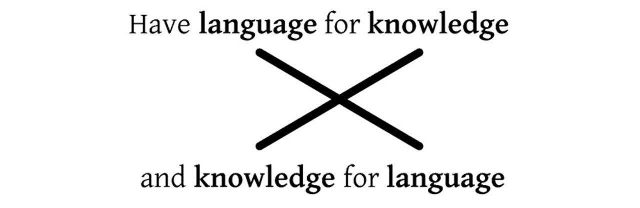
Have **language** for **knowledge**
and **knowledge** for **language**

FIGURE 1 | Schema of a chiasmus.

Epanaphora[1] is defined as the repetition of a word or a group of words at the beginning of successive sequences of language, where sequences can be defined in different ways. One can talk about epanaphora of chapters, lines, clauses or phrases. In this paper, we limit the scope to epanaphora of sentences, exemplified in Example 1.

(1) **I am** an actor.
I am a writer.
I am a producer.
I am a director.
I am a magician.

At the opposite end, epiphora[2] is the figure of speech of repetition at the end of a sequence (see Example 2).

(2) I'm so **gullible**.
I'm so damn **gullible**.
And I am so sick of me being **gullible**.

As for epanaphora, one can talk about epiphora of chapters, lines, clauses or phrases but we limit also the scope to epiphora of sentences, exemplified in Example 2.

We see different reasons why natural language processing should pay attention to figures of speech in general and to figures of repetition in particular. First, it may be useful for literary analysis: tools for supporting studies of literature exist but mostly belong to textometry. Thus, they mainly identify word frequency and some syntactic patterns, not figures of speech. Second, as shown in Dubremetz and Nivre (2015) those figures playing on repetition of words can be rare, or even extremely rare (sometimes only one figure is to be found in several hundred pages of books). Such rareness is a challenge for our discipline. NLP is accustomed to treating common linguistic phenomena (multiword expressions, anaphora, named entities), for which statistical models work well. We will see that chiasmus (and epanaphora/epiphora to a lesser extent) is a needle in the haystack problem. Thus we have a double-fold challenge: we must not only perform well at classifying the majority of spurious instances but above all perform well in finding the rare genuine cases.

The notion of a repetitive figure is vague. Dictionaries of stylistics tend to quote the same prototypical examples, which is not helpful when trying to capture the linguistic variety of them. The purpose of the linguists is to define each repetitive figure compared to other figures (for instance chiasmus as opposed to parallelism). To the best of our knowledge there is no pure linguistic study that tries to distinguish between, for instance, chiasmus and non-figure repetitions. In traditional linguistics, as opposed to computational linguistics, rhetorics is taken for granted. Linguistics has to answer only one question: Which figure is instantiated by this piece of rhetoric? Computational linguistics now has to answer not only this question but also the question of whether a piece of text is a piece of rhetoric in the first place. Repetition of words is an extremely banal phenomenon and we want to select only repetitions that constitute a figure of speech i.e., "a use of language that creates a literary effect[3]" or "an expression, [...] using words in an [...] unusual manner to add vividness, beauty, etc." (Neufeldt and Guralnik, 1997).

Gawryjolek (2009) was the first to address the automated detection of repetitive figures and of chiasmus in particular. Following the general definition of the figure, he proposed to extract every repetition of words that appear in a criss-cross pattern. His research shows that this pattern is extremely frequent while true chiasmi are rare. To give an idea of the rarity, in Dubremetz and Nivre (2015) we give the example of *River War* by Winston Churchill, a book consisting of 150,000 words, with 66,000 examples of criss-cross patterns but only one real chiasmus[4]. Hromada (2011) then proposed to add a feature constraint to the detection: he drastically reduced the number of false positives by requiring three pairs of words repeated in reverse order without any variation in the intervening material. However, in the example of Churchill's book, this also removes the one real example and the user is left with nothing else than a totally empty output. Finally, in Dubremetz and Nivre (2015) we built on the intuition of Hromada (2011) and added features to the detection of chiasmus, but in a different way. We observed that chiasmus, like metaphor (Dunn, 2013), can be seen as a graded phenomenon with prototypical examples and controversial/borderline cases. Thus, chiasmus detection should not be a binary classification task. Instead, we argue that a chiasmus detector should extract criss-cross patterns and rank them from prototypical chiasmi to less and less likely instances (Dubremetz and Nivre, 2015).

A serious methodological problem for the evaluation of chiasmus detection is the massive concentration of false positives (about 66,000 of them for only one true positive in 150,000 words). Such a low ratio makes the constitution of an exhaustively annotated corpus extremely time consuming and repetitive.

Because of lack of data, we tuned our features manually in Dubremetz and Nivre (2015, 2016). Those features included stopwords, conjunction detection, punctuation, position,

[1] In rhetorics, *epanaphora* is better known under the competing term *anaphora*. However, in computational linguistics, the term *anaphora* can be ambiguous as it refers as well to a referential pattern. For the sake of clarity, we will only use the term *epanaphora*.

[2] Epiphora is also known under the term *epistrophe*, but for consistency with *epanaphora* we will only use the term *epiphora*.

[3] Definition of "rhetorical device" given by Princeton wordnet: https://wordnet.princeton.edu/

[4] **Ambition** stirs **imagination** nearly as much as **imagination** excites **ambition**.

similarity of n-gram context, and syntactic role identity[5]. We could evaluate the hand-tuned system by average precision but it was only in Dubremetz and Nivre (2017) that we could make use of the annotations produced in earlier work to train a classifier using logistic regression. The latter system will be described in more detail in the experiments on chiasmus in section 3.

Epanaphora and epiphora have received even less interest from computational linguists. Actually, only one study on detection exists and it focuses on epanaphora. Strommer (2011) is the first to have applied machine learning to repetitive figures of speech. His underlying aim is to use epanaphora as a metric of genre. For this task, his detection needs to be as precise and exhaustive as possible. Thus, Strommer (2011) starts from a broader definition of epanaphora than we do: he accepts that some epanaphora could have sentence gaps as in Example 3.

(3) **I felt** moody and irritable.
I felt squished inside, I felt like standing in a field and twirling in circles [...].
Is it the driver's license?
I felt overwhelmed by it tonight.

This definition is acceptable, but makes the task even more complicated. Strommer reports technical difficulties mainly in getting enough annotations. Despite these difficulties, he describes some features useful for epanaphora and among them, some of them easy to transpose or use on epiphora detection as well. Those are the number of sentences, the presence of "strong" punctuation marks (! and ?) and the length of sentences (shorter than 10 words).

Epanaphora and epiphora thus have not attracted as much attention from computational linguists as chiasmus. Nevertheless the progress made on chiasmus through the very recent years might benefit the research on epanaphora and epiphora as well. And if so, this would support the idea that detecting figures of speech is possible even with the limited human resources that generally apply to figures of speech in general.

In this article, we will reuse the model introduced in Dubremetz and Nivre (2017) for the detection of chiasmus and generalize it to epanaphora and epiphora. We will start by presenting a generic approach to rhetorical figure detection, conceptualizing detection as a ranking task and giving a general characterization of evaluation methods, models and candidate extraction (section 2). We then present three concrete instantiations of this approach for, respectively, chiasmus, epanaphora and epiphora, trained and evaluated on data from Europarl (Koehn, 2005). The study on chiasmus, presented in section 3, has been previously published in Dubremetz and Nivre (2017), but the study on epanaphora and epiphora, in section 4, is original work presented for the first time. Finally, we will apply the three detectors in a case study on genre analysis, comparing the frequency of different figures in scientific titles, fiction titles, and quotations (section 5).

2. A GENERIC APPROACH TO RHETORICAL FIGURE DETECTION

Before addressing specific figures, we need to answer four questions common to any detection of repetitive figures. What is the task we are trying to solve? How do we evaluate the performance? What type of model do we use? How do we extract the candidates?

2.1. The Task

Even if someone could design the perfect detector that would output all and only the repetitions provoking a rhetorical figure, it is not certain that this would be the ideal system. As we observed in Dubremetz and Nivre (2015), chiasmus, like metaphor (Dunn, 2013), can be seen as a graded phenomenon with prototypical examples and controversial/borderline cases such as Examples 4, 5, 6.

(4) It is just as contrived to automatically allocate **Taiwan** to **China** as it was to allocate **China**'s territory to **Taiwan** in the past.

(5) "**We** have more to tell **you** than you have for **us**," said Phelps, reseating himself upon the couch.

(6) I **know** that every **word** is true, for you have hardly said a **word** which I did not **know**.

Thus, chiasmus detection needs not to be seen as a binary classification task. Instead, we argue that a chiasmus detector should extract criss-cross patterns and rank them from prototypical chiasmi to less and less likely instances. We believe this is true for other figures like epanaphora and epiphora as well. Indeed, it is easy to label as True the repetitions in successive sentences when those sentences are numerous, short and/or contain powerful repeated words as in Example 7. It is easy as well to label as False an instance that contains a single repetition, involves long sentences and rather neutral words like Example 8. However, there are also many cases in between that share properties of both true and false instances and that we cannot sharply place in one or the other category like Example 9.

(7) I consider this as **a disgrace**!
This is **a disgrace**!
A disgrace!

(8) We can accept the principle of prohibiting the exportation of Category 1 and Category 2 **material**.
We can agree to the principle of refrigeration of raw Category 3 **material**.

(9) So you want to give them a **national State.**
This is precisely what they need: another Syrian or Yemeni national State, a State of whatever kind; they need a **national State.**

[5] For a full description and justification of all features the reader can refer to Dubremetz and Nivre (2015, 2016)

The fact that controversial cases, like Examples 4 and 9, exist is not surprising and is not necessarily a problem in literature. Hammond et al. (2013) underlines that literature study is nourished by the plurality of interpretations of texts. The fact that Examples 4 and 9 can be interpreted as either a rhetorical figure or a non-figure repetition is interesting for a literature analyst. Thus, eliminating those examples would be an arbitrary choice made by the machine that would not help the plurality of interpretation desired by the humans. And, if overused, a detector with only a binary output could even create a bias toward the machine that would normalize the interpretation made out of repetition of words.

To solve this issue and make an effective detector that gives complete control to the literature analyst, we decide to see the task not as a binary task but as a ranking task. The machine should give all the instances of repetitions but in a sorted manner: from very prototypical true instances (like Example 7) to less and less likely instances. Thus the user would benefit from the help of the machine without completely losing their ability to choose which borderline case is useful for their literary interpretation.

2.2. The Evaluation

Redesigning the task into a ranking one was the easiest way to take into account the non-discrete property of the phenomena we search for. However, it makes the evaluation less straightforward. In an ideal world we would like to have a set of thousands of repetitions of each category (chiasmus, epanaphora, epiphora) all ranked by rhetorical effect power. Then we would try to achieve this exact ranking with a machine. The problem is that creating such a corpus would be very difficult and time consuming. Annotation time, given the noise generated by repetition extraction, is the real bottleneck of the detection problem. Besides the fact that it is very time-consuming to annotate all candidates, it is a very challenging task for an annotator to sort them into a complete ranking. As a practical compromise, we therefore limit annotation to three categories: True, False and Borderline. However, instead of evaluating only by precision and recall, we use average precision[6], which does not measure only binary decisions but whether true instances have been ranked higher than irrelevant cases. Moreover, when using data annotated by multiple annotators we count as True only those instances that have been annotated as True by all annotators. In this way we make sure that systems are evaluated with respect to their capacity to rank good, prototypical instances of a figure above all other. We consider this a reasonable compromise between the theoretical ideal of having a complete ranking of candidates and the practical necessity of making annotation and evaluation feasible. Finally, the fact that we have used a three-way categorization into True, False and Borderline makes it possible to later apply more fine grained evaluation methods[7].

While average precision, unlike precision and recall, is sensitive to the ranking of candidates, it nevertheless presupposes that we can identify which candidates to regard as True and False respectively. However, as noted earlier, it is practically impossible to exhaustively annotate all instances given the endless character of the problem. For instance, in a single book, there can be many thousands of candidates (Dubremetz and Nivre, 2015) for only one real chiasmus to be found. Luckily, treating the task as a ranking task helps us manage this problem as well. Here we seek inspiration from another field of computational linguistics: information retrieval targeted at the world wide web, because the web cannot be fully annotated and a very small percentage of the web pages is relevant to a given request. As described almost 20 years ago by Chowdhury (1999, p. 213), in such a situation, calculating the absolute recall is impossible. However, we can get a rough estimate of the recall by comparing different search engines. For instance Clarke and Willett (1997, p. 186), working with Altavista, Lycos and Excite, made a pool of relevant documents for a particular query by merging the top outputs of the three engines. We base our evaluation system on the same principle: through our experiments our different "chiasmus/epanaphora/epiphora retrieval engines" will return different hits. We annotate manually the top hundreds of those hits and obtain a pool of relevant (and irrelevant) repetitions. In this way, we can measure average precision in the top hundreds without having to do exhaustive annotation. In addition, we can measure recall, not absolutely but relative to the total pool of genuine cases found by all systems evaluated, as is commonly done in information retrieval evaluations.

2.3. The Model

In Dubremetz and Nivre (2015) we propose a standard linear model to rank candidate instances:

$$f(r) = \sum_{i=1}^{n} x_i \cdot w_i$$

where r is a candidate pattern, x_i is a set of feature values extracted from r, and w_i is the weight associated with feature x_i. Given candidates r_1 and r_2, $f(r_1) > f(r_2)$ means that r_1 is more likely to be a true figure of speech than r_2 according to the model.

We chose the linear model for its simplicity. As it just adds (weighted) features, a human can easily interpret the results. That allowed us in Dubremetz and Nivre (2015) to design detectors tuned manually when there was no data yet available for automatic tuning. Once we have accumulated enough training data, we can train a model using logistic regression

[6] Average precision is a common evaluation used in information retrieval. It considers the order in which each candidates is returned by making the average of the precision at each positive instance retrieved by the machine. Thus this measure gives more information on the performance of a ranking system than a single recall/precision value (Croft et al., 2010).Average precision is calculated on the basis of the top n results in the extracted list, where n includes all positions in the list until all relevant instances have been retrieved (Zhang and Zhang, 2009). The average precision is expressed by the following formula: $\sum_r \frac{P@r}{R}$
Where:
r = rank for each relevant instance
$P@r$ = precision at rank r
R = number of relevant instances in gold standard

[7] Although during training and evaluation, the borderlines are always counted as False instances, the borderline annotation is saved for future research and is already used to discuss the performance of our system in section 3.3.4.

(Pedregosa et al., 2011), which gives us a log-linear probability model, whichis a special case of the linear model where scores are normalized to form a probability distribution. Where scores are normalized by a probability distribution. This allows not only to do ranking (like we did with the human tuned system) but also to additionally give a precision and relative recall score, because every instance with a score above 0.5 is considered as a true instance by the model. Moreover, we can tune the probability threshold if we want to favor precision over recall or vice versa.

2.4. The Method of Extraction

Part of defining the task consists in choosing how we extract the candidates. There is not one obvious answer to this question.

The notion of repetition of identical words, common to the three repetitive figures, is ambiguous. "Identical" can refer to any type of identity, from vaguely synonymous to exact repetition of the same string. Ideally we should deal with all of them but in reality the task of extracting any kind of identity would pile up technical difficulties and make us extremely dependent on the performance of lexical resources available (stemmers, dictionaries, etc.). To make the tasks feasible we have to choose one method of extraction adapted to the resources we have and to the difficulties we are able to cope with. From a technical point of view, the most computer friendly methods of extraction are: matching the exact same string, as it has been done in previous work (Gawryjolek, 2009; Hromada, 2011), or matching the same lemmas as lemmatizers are now expected to be reasonably reliable in English.

Previous work (Hromada, 2011) has shown that we have to decide not only the type of repetitions we require, but also on the minimal number necessary. One can consider to extract only candidates that are based on more than a single repetition. It is a restriction of the definition, like restricting the kind of identity is, but it can be reasonable if it makes the task feasible.

So, how do we choose between these two parameters (extracting the same string vs. the same lemma, and requiring only one vs. several repetition of words). To answer this question, we perform a systematic exploration study that consists in extracting the candidates with a minimum of only one identical lemma, without any filter, and annotating a random sample of 100 candidates.

As we can see from **Table 1**, even if the method of extraction is the same (minimum one lemma repetition), the number of instances is definitely not the same, with chiasmus candidates being a thousand times more frequent than epiphora candidates. Additionally, the ratio between True and False instances is different. This calls for a close examination of the way to extract candidates. In the next sections we will see how we handle each figure in turn. For each of them, we will start by discussing the extraction method. We will go on to describe the features used in the respective models, and we will finish with experimental results based on the Europarl corpus.

TABLE 1 | Annotation of 100 randomly selected chiasmus, epanaphora and epiphora candidates.

Type of instance	True	Borderline	False	Number of candidates
Chiasmus	0	0	100	2,097,583
Epanaphora	1 ± 1.94	3 ± 3.33	96 ± 3.82	10,249
Epiphora	4 ± 3.77	7 ± 4.91	89 ± 6.02	2,723

The corpus is 4M words of parliamentarian discourses (159,056 sentences).

3. CHIASMUS

3.1. Extraction of Candidates

For chiasmus extraction, we extract every criss-cross pattern that has an identity of lemmas within a window of 30 tokens. Thirty tokens is the upper bound found empirically by Dubremetz (2013)[8] and reused by us in Dubremetz and Nivre (2015). As seen in **Table 1**, chiasmus is a pattern that generates an extremely large number of false instances (2 million instances, and 0 true instances in our sample of 100). Given the rarity of chiasmus and the absence of filters we are unlikely to find any true instance in the 100 randomly taken examples. Intuitively, we know that stopwords like articles, conjunction etc. are a factor of false instances. What we discovered during annotation of the 100 randomly taken instances is that it is even hard to find any other kind of false examples in such a small sample: all our chiasmus candidates involved the repetition of stopwords.

The number of true instances of chiasmus in the corpus is the most difficult to estimate. Proportionally our sample (100 for more than 2 million instances) is one thousand times less informative than for epiphora for instance (100 on nearly 3 thousands). If we look only at this table we can assume that in this corpus there is between 0 and 1% real chiasmus, i.e., from 0 to 20,000 instances of real chiasmi. That is why in Dubremetz and Nivre (2015) we take the example of a book written by Churchill where only one chiasmus was to be found and in the same conditions of extraction we got 66,000 instances. If our parliamentarians were to make as much chiasmus as Churchill in his book, in the 2 million instances corpus there would not be more than 40 instances of chiasmus. Of course, this estimate is to be used carefully, because there is no reasonable way to have a very close approximation. Nevertheless, with this comparison, we get at least the intuition that we definitely should not expect thousands of good examples in this politician discourse. In fact, we should probably not even expect several hundreds of them.

3.2. Model Features

Several features have already been tested for chiasmus, as discussed briefly in section 1. Below we list the features used in our model using the notation defined in **Figure 2**. The first set of features (1–17) are basic features concerned with size, similarity and lexical clues and come from Dubremetz and Nivre (2015), while a second group of features (18–22) belonging to the category of syntactic features was added in Dubremetz and Nivre (2016). We use the same features in our machine learning experiments but only train two systems, one

[8] In the corpus study of Dubremetz (2013) the largest chiasmus found consisted of 23 tokens.

$$\underbrace{\text{In prehistoric times}}_{C_{\text{Left}}}\ \underbrace{\textbf{women}}_{W_a}\ \underbrace{\text{resembled}}_{C_{ab}}\ \underbrace{\textbf{men}}_{W_b}\ \underbrace{\text{, and}}_{C_{bb}}\ \underbrace{\textbf{men}}_{W'_b}\ \underbrace{\text{resembled}}_{C_{ba}}\ \underbrace{\textbf{women}}_{W'_a}\ \underbrace{\ .\ }_{C_{\text{Right}}}$$

FIGURE 2 | Schematic representation of chiasmus, C stands for context, W for word.

corresponding to Dubremetz and Nivre (2015) (called Base) and one corresponding to Dubremetz and Nivre (2016) (called All features).

1. **Punctuation:** Number of sentence punctuation marks (".","!","?") and parentheses ("(",")") in C_{ab} and C_{ba}
2. **Weak punctuation:** Number of commas in C_{ab} and C_{ba}
3. **Central punctuation:** Number of strong punctuation marks and parentheses in C_{bb}
4. **Stopword *a*:** True if W_a is a stopword[9]
5. **Stopword *b*:** True if W_b is a stopword
6. **Words repeated:** Number of additional repetitions of W_a or W_b in the context
7. **Size difference:** Difference in number of tokens between C_{ab} and C_{ba}
8. **Size *bb*:** Number of tokens in C_{bb}
9. **Exact match:** True if C_{ab} and C_{ba} are identical
10. **Identical tokens:** Number of identical lemmatized tokens in C_{ab} and in C_{ba}
11. **Normalized identical tokens:** Same as previous one but normalized
12. **Identical bigrams:** Number of bigrams that are identical in C_{ab} and C_{ba}
13. **Identical trigrams:** Number of trigrams that are identical in C_{ab} and C_{ba}
14. **Identical left and central context:** Number of tokens that are identical in C_{Left} and C_{bb}
15. **Conjunction:** True if C_{bb} contains one of the conjunctions "and," "as," "because," "for," "yet," "nor," "so," "or," "but"
16. **Negation:** True if the chiasmus candidate contains one of the negative words "no," "not," "never," "nothing" (included in context left and right)
17. **To:** True if the expression "from …to" appears in the chiasmus candidate or "to" or "into" are repeated in C_{ab} and C_{ba} (included in context left and right)
18. **Identical tags:** True if W_a W_b W'_b W'_a all have the same part-of-speech tag
19. **Identical dependencies *a-b′*:** Number of incoming dependency types shared by W_a and W'_b
20. **Identical dependencies *b-a′*:** Same but for W_b and W'_a
21. **Identical dependencies *a-a′*:** Same but for W_a and W'_a
22. **Identical dependencies *b-b′*:** Same but for W_b and W'_b

3.3. Experiments

Following the setup in Dubremetz and Nivre (2017), we compare two models for chiasmus detection, one with only basic features (1–17) and one with all features.

3.3.1. Data, Annotation and Preprocessing

The data used in our experiments in this and the following section comes from the English section of Europarl (Koehn, 2005). It is a corpus common in natural language processing convenient for experimentation. The type of English contained in Europarl is generic enough that we think the system is likely to be applicable on many other genres, like novels. Europarl is the transcription of discussions in the European assembly. Most of the persons talking in it are politicians, some of them have well prepared speeches likely to contain the figures we are looking for. Finally it is a reasonably challenging corpus as the parliamentarian speech is full of repetitive structures like Examples 8 and 10 that are not necessary figures of speech. That makes it interesting to explore.

(10) Question No 62 by (H-0633/00):
Subject: Subsidies for **growing tobacco**
Tobacco growing in the European Union is subsidized to the tune of millions of euros per year while at the same time over half a million EU citizens die each year of diseases caused by tobacco.

The preprocessing consists in tokeninizing, lemmatizing, tagging and parsing the corpora with the Stanford CoreNLP (Manning et al., 2014). This allows the extraction of shallow and deep syntactic features. In each experiment involving an evaluation on test data the annotation task is systematically given to two different annotators. The annotation was done by the authors of this study. It is an expert annotation (as opposed to a crowdsourcing one). Both annotators have studied literature analysis but at different schools, in different languages, and at different times. It is interesting to see through this annotation whether two experts, not belonging to the same school, can agree on the interpretation of the candidate repetitions. In order to avoid any bias toward the machine, the instances to annotate are presented to the annotator in a randomized order that has nothing to do with the machine ranking output.

The training corpus is an extract of 4 million words from Europarl, containing 159,056 sentences. It is the same corpus used for generating **Table 1**. It contains 2,097,583 chiasmus candidates. Through our previous efforts in Dubremetz and Nivre (2015, 2016), 3,096 of these have been annotated by one annotator as True, False, Borderline, or Duplicate[10]. The True, Borderline and Duplicate instances were then re-annotated by a second annotator. There were 296 of them. Only instances labeled True by both annotators will be considered as true

[9] The list of stopwords is defined by the generic list made available in the snowball stemmer project: http://snowball.tartarus.org/algorithms/english/stop.txt

[10] For example, if the machine extracts both "**All** for **one**, **one** for **all**" and "**All for** one, one **for all**," the first is labeled True and the second Duplicate, even if both extracts cover a true chiasmus.

positives in our experiments (at both training and test time). This makes sure that both training and evaluation is based on the most prototypical true examples.

The test corpus is a different extract from Europarl, containing 2 million words and 78,712 sentences. It is used only in the final evaluation of the tuned models (sections 3.3.3 and 4.3.3) and it was used as a test set in previous research and thus already contains some annotated instances (Dubremetz and Nivre, 2016, 2017). It contains 1,057,631 chiasmus instances. For the test phase, two annotators were asked to annotate the top 200 instances of each system. In total, this produced 533 doubly annotated instances in our test set containing one million instances in total.

3.3.2. Training

We train a binary logistic regression classifier and use 2-fold cross-validation on the training to set the parameters[11]. To fit the system, we use the 31 instances labeled as True by both annotators as our positive examples. All other instances are labeled as False and thus considered as negative examples (even if most of them are actually unknown, because they were never encountered during the hand-tuning process).

We tried training on only annotated instances but the results were not satisfying. Normalizing features by the maximum values to get only 0 to 1 features deteriorated the result as well. We tried over-sampling by giving a weight of 1,000 to all true positive instances; this neither improved nor damaged the results. Finally, we tried support vector machines (SVM), with rbf and linear kernels, and obtained similar average precision scores as for logistic regression during training. When it comes to F-score, the SVM, unlike logistic regression, requires an over-sampling of true positives in order to perform as well as logistic regression. Otherwise, it converges to the majority baseline and classifies everything as false.

Based on these preliminary experiments, we decided to limit the final evaluation on the unseen test set to the logistic regression model, as its probability prediction allows us to rank chiasmi easily. For the linear logistic regression implementation we used scikit-learn (Pedregosa et al., 2011).

3.3.3. Evaluation

Table 2 shows that the model with basic features only achieves a quite respectable (average) precision but suffers with respect to recall. Adding the syntactic features further improves precision but also increases recall quite significantly. The best average precision achieved is 70.8, which indicates that the system is capable of ranking true instances high on average. For a comparison with the older hand-tuned system, we refer to Dubremetz and Nivre (2017).

TABLE 2 | Results for logistic regression model on chiasmus detection.

Model	Av. Precision	Precision	Recall	F1-score
Base	57.1	80.0	30.8	44.4
All features	**70.8**	**90**	**69.2**	**78.3**

Inter annotator agreement κ = 0.69.
Bold values indicate the most important differences between all features experiment and ablation of one feature experiments, in the ablation study.

3.3.4. Discussion

To cast further lights on the results, we performed an error analysis on the cross-validation experiments (run on the training set). In the all-features experiment, we encountered 4 false positives. Of these, 3 were actually annotated as Borderline by both annotators, and 1 was annotated as Borderline by one annotator and False by the other, which means that none of the false positives were considered False by both annotators. To illustrate some of the difficulties involved, we list 5 of the 31 positive instances in the training set (11–15), followed by the 3 borderline cases (16–18) and the 1 case of annotator disagreement (19).

Positive

(11) We do not believe that the **end** justifies the **means** but that the **means** prefigure the **end**.

(12) Do not **pick** the **winners** and let the **winners pick**.

(13) Europe has no problem converting **euros** into **research**, but has far greater difficulty converting **research** into **euros**.

(14) That it is not the **beginning** of the **end** but the **end** of the **beginning** for Parliament's rights.

(15) It is much better to bring **work** to **people** than to take **people** to **work**.

Borderline

(16) In parallel with the work on these practical aspects, a discussion is ongoing within the European Union on determining the mechanisms for participation both by EU Member **States** which are not members of **NATO** and by **NATO** countries which are not EU Member **States**.

(17) In that way, they of course become the **EU**' s representatives in the Member **States** instead of the Member **States**' representatives in the **EU**.

(18) If there is discrimination between a black person and a white person, or vice versa, for example if someone discriminates against a **white** Portuguese in favor of a **black** Portuguese, or against a **black** Portuguese in favor of a **white** Portuguese, this is clearly unlawful racism and should result in prosecution.

[11] Since we had a very small number of positive instances, using 10-fold cross-validation would have made the validation procedure unreliable, so we instead opted for a simpler 2-fold cross-validation, using half of the data for training and the other half for validation. In order to avoid over-fitting, we repeated the process 6 times with different randomizations and used the average of these runs as the validation score.

Disagreement

(19) European consciousness is that which must contribute to the development of mutual respect [...] and which must ensure that tolerance is not confused with laxity and an absence of **rules** and **laws** and that **laws** and **rules** are not made with the intention of protecting some and not others.

How can the classifier achieve such good results on both recall and precision with only 31 positive instances to learn from? We believe an important part of the explanation lies in the way the training set was constructed through repeated testing of hand-crafted features and weights. This process resulted in the annotation of more than 3,000 obvious false positive cases that were recurrently coming up in the hand-tuning experiments. Our way to proceed consisted in first tuning the weights of the features manually. In this process we started by using the stopwords as our first feature in order to filter out most false positives. This is in fact a necessary requirement. Without stop word filtering, the chance of finding a true positive in the top 200 instances is extremely small. Thus, if a false negative is hidden somewhere in the training set, it is likely to be one involving stop words. To the best of our knowledge, there is only one existing chiasmus ever reported in the history of rhetorics that relies exclusively on stopwords[12].

Given this, we cannot guarantee that there are no false negatives in the training set, but we can definitely say that they are unlikely to be prototypical chiasmi. Thanks to this quality of the annotation, the machine had the maximum of information we could possibly give about false positives which is by far the most important class. In addition, the performance observed with only 31 positive training instances might be revealing something about chiasmus: the linguistic variation is limited. Thus, within 31 examples the patterns are repeated often enough so that a machine can learn to detect them.

We have now addressed the problem of chiasmus and discovered that even with a very partial annotation we can train a system. In the next section, we will generalize this approach to epanaphora and epiphora, two figures that have hardly been explored at all in computational linguistics.

4. EPANAPHORA AND EPIPHORA

4.1. Extraction of Candidates

As for chiasmus we perform the basic exploration of the epanaphora and epiphora patterns (**Table 1**) in order to determine the best extraction process: only successive sentences, with one identical initial lemma for epanaphora, final lemma for epiphora, are considered as candidates. Then 100 examples are randomly picked and annotated.

The epanaphora and epiphora extraction is definitely less of a needle in the haystack problem than chiasmus. As we can see in **Table 1**, the number of candidates is reduced to a couple of thousands instead of millions and we find at least one positive example and several borderline cases in our extraction of epanaphora and epiphora. That is extremely positive because it means that, unlike for chiasmus, we might not have to start tuning systems manually: a couple of state of the art filters should be enough to extract a decent number of positive examples in order to directly train our system.

We just showed that both epanaphora and epiphora extraction are less noisy than chiasmus extraction. The issue is now to determine if epanaphora and epiphora detection are really the same problem and thus could be extracted with the same parameters (kind and number of repetitions).

From **Table 1**, we see that the distribution of instances for epanaphora vs. epiphora is definitely not the same. First of all, the number of epanaphora candidates is more than three times larger than the number of epiphora candidates. During the annotation of epanaphora, we noticed the following: more than 50% (55 exactly) of the candidates are simply due to the determiner *The* occurring at the beginning of the sentences and 20% are due to the appearance of a single pronoun (*I*, *It*, *You*, etc.). Such recurrent patterns do not appear in epiphora candidates. Thus, epanaphora extraction is necessarily more noisy, which is confirmed by the number of true and borderline cases found in the samples: 11 True or Borderline cases found for epiphora and only 4 for epanaphora. Even if epanaphora detection seems more difficult than epiphora detection because of the number of candidates this can be handled like we do for chiasmus. However, another phenomenon attracted our attention and forced us to restrict the candidate selection for epanaphora: the phenomenon of True/False cases. In our machine learning system we want to divide the candidates into three categories: True like Example 7, False like Example 8, and Borderline like Example 9. Because of pronouns and determiners, we observed that epanaphora, more than epiphora and chiasmus, could generate instances that cannot really be defined as Borderline cases because they contain very prototypical True cases and very prototypical False cases at the same time. We observe this in Example 20. In this example, the fact that the author insists four times on the formulation *He should never have* is a noticeable rhetorical effect that would deserve to appear in a translation, or be stressed in a text-to-speech application. But the fact that the first sentence starts with the pronoun *He* is nothing exceptional and it would sound strange to stress it as a rhetorical figure.

(20) **He** arrived from Algeria at a time when, [...] immigration had been stopped and there was no reason for him to come.
He should never have stayed in France, having been reported to the transport police more than 40 times for offences [...].
He should never have been free because 14 charges of theft, violence and rape had been brought against him[...].
He should never have been able to escape from France, but the officers pursuing him had no jurisdiction and European frontiers have more holes than a sieve.
He should never have arrived in Spain, where he mugged a woman at knifepoint [...].

This phenomenon is confusing even for a human and can make the task of annotation and learning extremely difficult. Furthermore, we observed that in the case of personal pronouns

[12] **All** for **one**, **one** for **all**.

and verbs the lemmatizer was extracting candidates that did not even sound like a reasonable candidate to extract. For instance in the case of Example 21 the verb "be" is starting the sentence but has a very different morphology which makes the example likely to be considered as false. In Example 22, we see that the lemmatizer is tailored to lemmatize the same way subject pronouns and possessive pronoun. In this particular case the example would sound like a perfect positive one if only the machine had not extracted the last sentence, starting by "his." Such case is not so infrequent in the task of epanaphora detection. Indeed, personal pronouns (I, my, you, our, he, his...) are likely to be at the very beginning of sentences. The personal pronoun category is a closed class of words with only a dozen of possibilities, thus reducing even more this class by lemmatization sounds like an unnecessary factor of false candidates and/or of True/False cases like Example 22 was.

(21) **Are the** rights guaranteed under the Convention on Human Rights better than those guaranteed under the EU' s Charter?
Is the latter, again, better than the national constitutions?

(22) **He knows perfectly well that** ours is a non-political Head of State.
He knows perfectly well that for nearly fifty years she has scrupulously avoided engaging in controversial political issues.
He knows perfectly well that she cannot come to this House to set the record straight.
His behaviour is a disgrace and a scandal.

Because of these two phenomena, we decided to restrict the extraction to epanaphora candidates that have at least 2 identical words (not lemmas) at the beginning. Thanks to this restriction, the number of candidates is comparable to those for epiphora detection: 2,369.

This exploration determined for us the way to proceed in the extraction of candidates. It cast light on what method of extraction is preferable for each figure. In the specific case of epanaphora, using the same extraction method would lead to different numbers of candidates (more than three times more for epanaphora than for epiphora) and the types of false positives would not be the same. English grammar imposes different constraints at the beginning and at the end of a sentence. We therefore have to use two different extractions for the two problems.

In this section we have explored the common problems concerning both epiphora and epanaphora. In the following parts we present the features experiments and the specific implementation for each of these figures.

4.2. Model Features

In this section, we describe eight features that are used in our systems and that provide the basis for a feature ablation study in the next subsection. The three first features (baseline ones) are inspired by the previous study of Strommer (2011); the five others come from our own exploratory study.

1. **Sentence count:** As noted by Strommer (2011), the number of sentences exhibiting a repetition is a significant feature. The higher the number, the more likely the repetition is to have a rhetorical effect. We treat this as a simple numerical feature. For instance, Example 23 has a sentence count of 3. Sentence count is included in the baseline models.

(23) **Paranoid**?
Paranoid!
Who says I'm **Paranoid**?

2. **Strong punctuation:** The strong punctuation feature counts the number of sentences that end with a "strong" punctuation mark (! or ?). The count is normalized by taking the average over all sentences in the sequence. For instance, Example 23 has a strong punctuation feature of 3/3 = 1. Strong punctuation is also a baseline feature.
3. **Sentence length:** The third feature inspired by Strommer (2011) is sentence length, which measures the average number of tokens per sentence in the sequence. For instance, Example 24 has a sentence length feature of (6+5)/2 = 5.5. Sentence length is the third and final baseline feature.
4. **End similarity:** The end similarity feature counts the number of successive identical lemmas at the end of adjacent sentences, averaging over all such pairs in the sequence. To avoid giving an excessive advantage to long sentences, we divide this number by the number of words of the shortest sentence. For instance, Example 24 has an end similarity of 2 similar end words (*not on*) divided by 4 (the number of words in the shorter sentence *This is not on*).

(24) This is **not on**!
This is absolutely **not on**!

5. **Start similarity:** The start similarity feature is analogous to the end similarity feature but at the start of sentences.
6. **End tag similarity:** The end tag similarity feature is again analogous to the end similarity feature but looks at part-of-speech tags instead of lemmas. For instance, Example 25 has an end similarity score of only 1/3 = 0.33 but an end tag similarity score of 3/3 = 1. (The sequence of tags is identical for both sentences: pronoun, verb, pronoun.)

(25) I made **it**!
You take **it**!

7. **Same strict:** The same strict feature is a binary feature that is 1 if the last word of the sentences in a sequence has the same form as well as the same lemma. For instance, Example 27 has a same strict value of 1, while Example 26 has a same strict value of 0, because *problem* is repeated without the inflection *-s* the second time.

(26) As such, mining waste is one of our **major problems**.
The safety of mines is also a **major problem**.

8. **Diff on end similarity:** Diff on end similarity (DoE) is our most complex feature. It counts the number of identical lemmas at the end of sentences but then divides it by the

number of lemmas that do not reappear in the other sentence. For instance, Example 27 has two different words (*so*, *now*) but have only one identical word at the end of the sentence. Thus, the diff on end similarity is 2/1 = 2.

(27) And so **what**? And now **what**?

When this feature is applied to epanaphora, we call it diff on start similarity (DoS) because we then divide the difference by the n-gram similarity at the start of the sentences instead of the end.

4.3. Experiments

For epanaphora and epiphora the questions are slightly different than for chiasmus. First no ranking method has ever been tested for those two figures. It is a pattern that generates fewer false candidates than chiasmus, but only some features have been tested so far and only on epanaphora. Finally, unlike chiasmus, epanaphora and epiphora, have strong theoretical similarities as they are often defined together. The most legitimate question to answer is thus whether this theoretical proximity is confirmed in practice by testing the same set of features on the two figures.

4.3.1. Data, Annotation and Preprocessing

The corpora used for experiments in this section are the same as in section 3.3.1. We use the same preprocessing as we used for chiasmus and the same approach to annotation.

The training corpus is the same as in section 3. With our method of extraction (see section 4.1), this 4 million words training corpus contains 2,723 epiphora candidates and 2,369 epanaphora. Annotation is time consuming, thus not all of these instances are annotated. To make the task feasible, we annotate only the candidates preselected by any of the features of Strommer (2011). All other candidates will be assumed to be negative instances (i.e., candidates that neither have any strong punctuation, neither more than 2 sentences or do not have an average of less than 10 words per sentences.). For instance, Examples 7 and 9 would be annotated but not Examples 8 and 28.

(28) **It is** not exactly the first successful conciliation on social matters between the European Parliament and the Council. **It is** the second, following the successful conciliation on the minor issue of workers working in an explosive atmosphere.

This first round of annotation represent in total a set of 508 epanaphora candidates and 410 epiphora candidates. These were first annotated once by one annotator. Then all instances annotated as True or Borderline were sent to a second annotator and discussed. Only the candidates considered as True by both annotators were used as True instances for training (64 True epanaphora instances, 50 True epiphora instances). All remaining instances were regarded as False (even though most of them were actually unknown because they were never encountered during any of the annotation process).

The test corpus is also the same as in section 3.3.1. It contains 1,154 epanaphora instances and 1,164 epiphora instances. It is used only in the final evaluation of the tuned models (with only the top 200 instances of each systems annotated, as described in section 2.2). This evaluation method yielded to the annotation of 291 epiphora candidates and 297 epanaphora candidates (among them 35 epiphora and 53 epanaphora were doubly annotated as True instances).

4.3.2. Training and Feature Selection

To test the usefulness of our features for detecting epanaphora and epiphora, respectively, we performed an ablation study, where we systematically removed one feature at a time to see what contribution it gave to the results. Based on the result of the ablation study, we then tried to select the best model for each figure of speech. All the feature selection experiments reported in this section were performed on the training corpus.

The feature ablation study was carried out by training and evaluating a binary logistic regression classifier using two-fold cross-validation (Pedregosa et al., 2011). This is essentially the same set-up as for section 3. The only difference is that we applied oversampling with a weight of 5 for the true class. We tried not oversampling at all, but this degraded the F-score because of a recall lower than 10%.

Table 3 shows the results of feature ablation for epanaphora. Two results are noteworthy. First, sentence length (abbreviated as Length), unlike other basic features, does not seem to make a positive contribution to the result, as seen by the fact that

TABLE 3 | Ablation study for epanaphora.

Epanaphora	F-Score Av. P.	Δ Full feat.
Full Features	62.25% 54.60%	– –
-Sent. Count	48.59% 54.02%	**–13.66** –0.58
-Strong Punct.	56.37% 50.30%	**–5.88** –4.30
-Length	63.13% 55.71%	**0.87** **1.10**
-Start Sim	61.12% 54.51%	–1.14 –0.09
-End Sim.	62.01% 54.59%	-0.25 –0.02
-Start Sim Tag	62.03% 54.65%	–0.22 0.05
-Same Strict	N/A (see section 4)	
-Diff on Start	48.29% 42.58%	**–13.96** **–12.03**

Bold values indicate the most important differences between all features experiment and ablation of one feature experiments, in the ablation study.

accuracy improves when this feature is removed. Secondly, one feature seems to be much more important than all others, namely diff on start (DoS), since results drop by over 10 points when this feature is removed, which is ten times more than the second best new feature, start similarity, where results drop by slightly more than 1 point.

Based on the results of the ablation study, we decided to run an additional model selection experiment, the results of which are shown in **Table 4**. Here we see that a simple combination of the baseline and the DoS feature performs almost as well as the full feature model (less than 1% difference). And if we remove the harmful sentence length feature, it actually performs even better (gain of 1% on both metrics compared to Full Features). We therefore selected Baseline − Length + DoS as the final model to evaluate on the test set.

The ablation study for epiphora, shown in **Table 5**, tells a different story. Here all the features seem to make a positive contribution, and no feature stands out as remarkably better than any other, although the two baseline features strong punctuation and sentence length appear to be crucial for getting high average precision. As a sanity check, we also made a model selection experiment including a model with only baseline feature and diff on end (the counterpart of diff on start, which was so important for epanaphora). However, the results in **Table 6** confirm that, for epiphora, the full model is indeed the best performing model when using cross-validation on the training set (plus 3 points for full features on both F-Score and average precision compared to Baseline + Diff on End experiment). We therefore selected this model to be evaluated on the test set.

4.3.3. Evaluation

Tables 7, 8 shows the evaluation results for the baseline and the two best models on the unseen test set. The results were obtained by annotating the union of the top 200 instances output by the four systems as proposed in Dubremetz and Nivre (2015) with inspiration from Clarke and Willett (1997). We observe that the best models improve on all metrics by at least 14%. The improvements are balanced across recall and precision and end up improving the F-Score by 20% for both figures. The largest improvement is obtained in the average precision of epanaphora (+38%). This difference is actually the most impressive because it is created only by the addition of the DoS feature and the removal of the sentence length feature. Like the baseline model, the best epanaphora model has only three features, and yet improves the F-score by 24%.

TABLE 4 | Choosing the best model for epanaphora.

Epanaphora	F-Score	Δ Baseline
	Av. P.	
Baseline	35.96%	–
	31.74%	–
Full Features	62.25%	+26.29
	54.60%	+22.86
Baseline + DoS	61.18%	+25.22
	54.75%	+23.01
Baseline - Length + DoS	63.73%	**+27.77**
	55.63%	**+23.89**

Bold values indicate the most important differences between baselines and experiments.

4.3.4. Discussion

How can we explain that for epanaphora, unlike epiphora, only one new feature is needed to significantly improve the results?

TABLE 5 | Ablation study for epiphora.

Epiphora	F-Score	Δ Full feat.
	Av. P.	
Full Features	51.80%	–
	60.53%	–
-Sent. Count	51.28%	−0.52
	60.37%	−0.16
-Strong Punct.	49.89%	−1.91
	52.90%	−7.63
-Length	51.13%	-0.67
	51.15%	**−9.39**
-End Sim.	51.41%	−0.39
	60.27%	−0.26
-Start Sim.	50.74%	−1.06
	59.13%	−1.40
-End Sim. Tag	50.64%	−1.16
	59.66%	−0.87
-Same Strict	50.50%	−1.30
	59.38%	−1.15
-Diff On End	49.28%	−2.52
	58.58%	−1.95

TABLE 6 | Choosing the best model for epiphora.

Epiphora	F-Score	Δ Baseline
	Av. P.	
Baseline	35.11%	–
	41.91%	–
Full Features	51.80%	**+16.69**
	60.53%	**+18.62**
Baseline + DoE	48.48%	+13.37
	56.29%	+14.38

Bold values indicate the most important differences between baselines and experiments.

TABLE 7 | Results for the epiphora experiments.

Experiment	Recall	Precision	F-Score	Av. Prec.
Baseline	25.71	42.86	32.14	26.78
Full Features	45.71	64.00	53.33	47.90
Δ	+20	+21	+21	+21

Inter annotator agreement Cohen's $\kappa = 0.88$

TABLE 8 | Results for the epanaphora experiments.

Experiment	Recall	Precision	F-Score	Av. Prec.
Baseline	30.19	29.09	29.63	19.97
Baseline -Length + DoS	45.28	53.33	48.97	57.92
Δ	+15	+14	+24	+38

Inter annotator agreement Cohen's $\kappa = 0.85$.

How can we explain that DoS (and DoE to a lesser extent) is such an effective feature? There could be several reasons for this.

First, the order in which we performed experiments may have played a role. We started by designing and testing features on epiphora and then adapted these to epanaphora. We did not add or test new features especially designed for epanaphora. This may partly explain why we ended up with a simpler system for epanaphora. If we had reversed the order, we might have ended up with fewer features applied to epiphora.

A second possible reason is the way we extract candidates, which is subtly different for the two figures. Because of the high number of false positives beginning with a single repeated function word for epanaphora, we had to require at least two repeated words, which may have reduced the effectiveness of some features like similarity of beginning.

These reasons may explain why there are fewer features in the best epanaphora model. However, they do not explain why DoS is such a powerful feature. One explanation may be that this feature, unlike other features, combines two properties: similarity and difference. Indeed, before coming up with this feature we tried using a simpler measure of the difference, without normalizing by the length of the repetition. This was helpful, but not as good as the normalized version, and it turned out to be redundant and harmful when used together with the normalized version. DoS is the only feature that measures the relation between two properties: similarity vs. difference. Without measuring this relation between the two features we probably miss an essential property of the figure.

Finally, this might be explained by the definition of the figure itself. Our inter-annotator agreement is good but it was achieved after discussions between annotators on many borderline cases in the exploratory study. Before this discussion, our inter-annotator agreement was below 40% for both of our figures. What came out of discussions is that the rhetorical effect of epanaphora and epiphora often comes from the combination with another figure of speech. For instance, Example 29 contains rhetorical questions, Example 30 contains a parallelism, and Example 31 is an apostrophe.

(29) Are the profits from the arms trade **clean money**?
Are the huge sums spent bribing officials [...] **clean money**?
Are the profits amassed by [...] companies by making children work [...] **clean money**?

(30) National states provide development aid**, so does Europe**.
National states combat racism**, so does Europe**.
National states support the women' s movement**, so does Europe**.
National states support the trade unions and parties**, so does Europe**.

(31) **Poor old** European Commission!
Poor old European Council of Ministers!
Poor old European Union!

Others may trade on hyperboles, allusions or other figures. This means that there are many different ways to build the rhetorical effect of repetitive figures and every human is not equally equipped to perceive that. Some annotators are more sensitive to the similarity of syntactic structures, others are more disposed to attend to lexical phenomena. Human annotators can have a difference of sensitivity toward rhetorical effects. We can illustrate this by Example 32: in this particular case, one annotator could see the allusion to the similar expression "the poor cousin," while the other one could not see it, because he did not know the expression.

(32) **It has been the poor relation** with respect to the milk, beef, tobacco and wine sectors.
It has been the poor relation because it has been deemed to lack a strong voice in Europe.

This is the proof that difference of education modifies our perception. However, there is something common to all humans: we all have a limited memory. Thus, the more memorable a pair of sentences is (more repetition between them and less differences), the more likely they are to be perceived as rhetorical by all humans, regardless of their backgrounds. Thus, DoS and DoE features work because they encode a more universally perceived property.

Now we have designed systems of extraction for three figures of speech. They have been trained and tested on data from Europarl. In the next section, we will see if our systems prove to be useful on other corpora.

5. A CASE STUDY IN GENRE ANALYSIS

In the preceding sections we developed three systems of detection for three different figures of repetition. This is the first time that detection of such a large set of repetitive figures has been both developed and fully evaluated. As we know our systems are not able to give an absolute recall. However, this does not impair the quantitative analysis capacity as long as we create a fair comparative study. In this section, we will apply the three detectors to three comparable corpora (same quantity of text, same language and only different genres). We will study how the genre influences the frequency of different repetitive figures.

While studying chiasmus, one remark attracted our attention. Vandendorpe (1991, p.4) says:

> Very commonly used in the 70's, [chiasmus] has been harshly criticized for the violence it makes against communicative function of language : "[...]. The research of meaning is the meaning of research, etc. You can appear deep with any banal idea." But by forcing the naive reader to think, this propositional chiasmus [...] is often appropriate for titles both because of its lexical economy and because of the endlessly deep discussions that it seems to foretell. In *La trouble-fête*, Bernard Andrès criticizes this process for being typical of academic jargon[...], which is likely to draw attention and grants.

The remark of Vandendorpe (1991) citing Bernard Andrès implies several assumptions. First, if chiasmus is convenient for titles, we might be likely to find them in this kind of text. Second, chiasmus seems to be regarded by Vandendorpe as cliché[13] when it comes to academic writing. This remark is interesting because it assumes that chiasmus should be over-frequent in it. However this was said nearly 30 years ago. There is no way that Vandendorpe (1991) could check automatically on huge amounts of academical titles that chiasmus is a cliché in this genre. We have no reason to disbelieve what Vandendorpe (1991) says but it is discussable. One could argue that chiasmus, and any figure of speech in general, should be less frequent in scientific titles given that researchers are not professional creative writers like authors of fictions are.

To check this, we design a study on multiple corpora. The first corpus contains titles from scientific publications, and the second contains titles from fictional literature. Running our detectors on both of those corpora is interesting because they are alike (they are both a list of titles) and they come from completely different genres. To improve our comparisons, we also include a corpus of quotes[14]. The quote corpus is likely to contain rhetorical figures, because quotes are selected for their rhetorical properties. Thus, in quotes, figures of speech are expected, more than in any other genres and certainly more than in titles in general. If a figure of speech in scientific articles happen to appear nearly as much as in quotes we can definitely conclude that this figure is a cliché.

5.1. Data, Annotation and Preprocessing

The three corpora[15] we use are:

- **The Fiction Titles Corpus**: It is obtained by downloading all the titles of books under the category "Fictions" available on the website Waterstone[16]. After, cleaning and removing of duplicates this corpus contains exactly 192.506 titles.
- **The Scientific Titles Corpus**: We download titles from scientific publications coming from dblp[17]. For comparison sake we apply the same preprocessing as for the fiction titles corpus and we reduce the number of title by picking randomly 192.506 of them.
- **The Quotes Corpus**: We download a corpus of quotes. We have 32,000 quotes but with a comparable number of words with the fiction title corpus (around 1,000,000 words).

Out of those three corpora, two are lists of titles. Titles are suitable for comparing genres because a title is an independent meaningful piece of text and it is easy to obtain corpora of equivalent sizes simply by sampling the same number of titles. Titles are also short, which means that the number of epanaphora and epiphora candidates is limited. This allows us to exhaustively check all candidates manually, which is otherwise difficult.

Those three corpora are used in a realistic condition of a user in order to perform a literary analysis of the genres. At this stage of the study we want to reproduce the same conditions as a real discourse analyst would face. Thus, the corpora are annotated by one annotator without randomization. The corpora are preprocessed (tagged and parsed) as described in previous sections before running the detectors.

5.2. Experiments

The results of our comparison are presented in **Table 9**, where we report three types of results. The first is the number of true positive instances found with a probability threshold of 0.5 (@S = 0.5). This is useful because it allows to compare the number of figures in each corpus on an equal basis, and 0.5 is the natural threshold above which a classifier considers an instance as true. However, outside of the quote corpus, very few instances have such a high score and the user is likely to look below this score. Therefore, we also give the number of true instances at rank 100 or above (@R = 100) (or at the maximum rank when there are too few candidates). Finally, in the third column (in blue), we evaluate the system using average precision. The results show that average precision can be excellent even when very few true instances are to be found. When the average precision reaches 100%, this means that the user never has to encounter any false positives because all true instances are ranked first.

Reading **Table 9** and comparing it to **Tables 2, 7**, **8**, the reader might be struck by the strong performance of the system (over 90% precision for the majority of the figures and corpora). The average precision scores are higher than in our development corpus. On the one hand, this improvement could be expected because the task itself is easier as titles and quotes, unlike Europarl, are already split into chunks of little texts. This leads to fewer false candidates thanks to the limited amount of long successions of non-figure repetitions. On the other hand, the system has never been trained on any of these corpora, and could thus have suffered from bad performance because of it. That means that Europarl was generic enough that our algorithm could be applied to those three different types of texts. The only drawback is that the chiasmus system achieves very high precision at the expense of recall. The probability

[13] i.e., expression that has lost originality, ingenuity, and impact by long overuse.

[14] The list of quotations comes from an open source collaborative collection initiated by Tan (2015).

[15] The three corpora and output based on them are available at https://github.com/mardub1635/corpus-rhetoric (Dubremetz, 2018).

[16] Waterstone is a commercial website for selling books to the general public https://www.waterstones.com

[17] dblp is a database of scientific publications in the domain of computer science http://dblp.uni-trier.de

TABLE 9 | Application of our best detectors on three corpora.

	Chiasmus			Epiphora (incl. Symploce)[18]			Epanaphora).		
	@S=0.5	**@R=100**	**Av. P. (%)**	**@S=0.5**	**@R=100**	**Av. P. (%)**	**@S=0.5**	**@R=100**	**Av. P. (%)**
Fiction	1/1	13/100	66	35/41	37/43	100	2/2	2/4	100
Scient.	7/7	21/100	65	3/4	3/13	100	0/0	1/2	100
Quotes	46/46	93/100	99	187/197	93/100	91	128/131	98/100	99

We report number of true instances retrieved with probability threshold 0.5 (@S = 0.5), number of true instances in the top 100 (@R = 100), and average precision (Av. P.).

threshold here is not very well calibrated, and the system hardly assigns a probability above 0.5 to any instance. For example, only one instance of chiasmus has a score above 0.5 in the fiction corpus. This, however, is not a problem for the user, because we do not limit the output to those. The ranking is what matters the most for the user and it is reflected by the average precision measure which is still excellent (minimum 65%).

Because titles are extremely short, the number of candidates is fairly limited for epanaphora and epiphora (only 2 instances of epanaphora for scientific titles for instance). This allows us to look at all the candidates which is excellent for corpus analysis. In the case of epiphora, we can say that our human annotation is exhaustive. Indeed, in the case of titles of fiction and titles of science, there were very few candidates (43 for fiction and 13 for science at the maximum) and unlike epanaphora the constraint for extracting candidates is minimal (only one repetition of lemma needed).

Our genre analysis confirms the intuition of Vandendorpe (1991). Chiasmus is nearly twice as frequent in scientific titles as in fiction titles, if we look at the top hundred. And if we limit our comparison to the the very prototypical instances scored over 50% we have seven times more of them. That can be explained by the fact that chiasmus is the ideal figure to wrap up an argument and above all to summarize a paradoxical issue as in Example 33 and 34.

(33) Doing the **Right Thing** or Doing the **Thing Right**: Allocating Resources Between Marketing Research and Manufacturing.

(34) A **Future** with No **History** Meets a **History** with No **Future**: How Much Do We Need to Know About Digital Preservation.

The fact that chiasmus is more frequent could be seen as normal because titles of science are longer than of literature. (On average, scientific titles contained twice as many words as literary ones.) If this was true, we should find more repetitive figures in general: this is not the case. Epanaphora and epiphora are almost non-existent in scientific titles (1 anaphora and 3 epiphora found) and we find nearly ten times more of them in titles of fiction (2 epanaphora and 37 epiphora and symploce).

Why are epanaphora and epiphora more specifically used in fiction title? Our explanation is based on the observation of the figures. Note that the misbalance in the number of epiphora is due to the majority of symploce found (30 of them in the corpus of fiction). Most of them are actually the short and lyrical repetition of one or two words like in Examples 35 and 36 extracted from two titles of thrillers. These short symploce are excellent in fiction titles because they are appealing, fast to parse for the reader and they give them an immediate emotion likely to make them open and buy the book.

(35) **Stop It**!
Stop It!
Stop It!

(36) **Bingo**!
Bingo!
Bingo!

In science the aim is different and emotional appeal is not enough to make a good title in this genre. The aim of the scientist is not only to be read but above all to be cited. To be cited, a scientist must show that he provides useful content to the scientific community. Epanaphora and epiphora, unlike chiasmus, takes a lot of words to express the content of an argument. For instance, in Title 37 and 38 the repetitions alone take 8 words in the title. If they are too short, they are emotionally appealing but they do not reveal what is the article content, problem or argument. For instance, Example 39 is a title that is appealing but does not precisely express which scientific domain the article belongs to. Thus, it might not reach the right audience.

(37) **Bring out your codes! Bring out your codes**! (Increasing Software Visibility and Re-use)

(38) **Beneath the layers in** nature, resilient life. **Beneath the layers in** artifacts, lifeless components.

(39) **Models**. **Models**. **Models**. So what?

Thanks to this case study, we show evidences that the origin of the text (fiction vs. science vs. quotes) influences the type of repetitions we are likely to find. Our case study supports, in a systematic way, the intuition of Vandendorpe (1991). We

[18] Sometimes both an epanaphora and an epiphora are contained in a repetitive figure either because it repeats the all sentence (e.g., **Torah**! **Torah**! **Torah**!) or because it is repeating the beginning and the end (e.g., **Life is a** song - sing **it**. **Life is a** game - play **it**. **Life is a** challenge - meet **it**.) this phenomenon is called symploce and in order to not count them twice we count them only as epiphora.

have showed, for instance, that chiasmus is more used in scientific titles than in fiction. However, we give a more nuanced judgement than he does: yes, chiasmus is more frequent in science titles than in fiction but not to the extreme that we meet in quotes. Finally, if we just confirm the intuition that quotes are full of figures of speech, our system allowed us to discover a more surprising result: the specificity of epanaphora, epiphora, and above all symploce to fiction titles.

6. CONCLUSION

In this article, we have targeted the detection through ranking of three repetitive figures: chiasmus, epiphora and epanaphora. The challenge consists in training a model for rare stylistic phenomena, with a corpus that is only very partially annotated. We have proposed a generic approach to this problem, using a linear model for ranking and average precision for evaluation, and we have shown that the model can be successfully applied to three different repetitive figures, each with its own characteristics. Finally, we have demonstrated the usefulness of our approach through a comparative analysis of three corpora: one belonging to science, another to fiction, and a last one consisting of quotes. In this way, we discovered that chiasmus was more specific to scientific publication titles, whereas epiphora and epanaphora were more likely to appear in fiction titles. This study is unique: for the first time the frequency of figures are compared mechanically on comparable corpora and we could detect the specificity of figures to different genre.

Such a tool and comparative method opens up to new type of literary analysis adapted to our century: in the recent past, the life of a literature scholar consisted in knowing and reading maybe a couple of hundred of canonical authors (Shakespeare, Fitzgerald, ...) already selected through the ages and through editing processed. In the internet century, the authors are millions. They write amateur books[19], short stories, poetry, blogs, etc. Some of them are talented and would deserve to be studied, but the overwhelming number of texts available makes it difficult to find them. Our tool is complementary to the traditional manual analysis. Thanks to our ranking system, we never pretend to replace the human judgment with a binary system. As expressed very well by Michael Ullyot, we do not aim at building robots, but we enable readers to become augmented[20].

AUTHOR'S NOTE

Section 3 is based on work previously published in Dubremetz and Nivre (2017).

AUTHOR CONTRIBUTIONS

MD has lead the experiments and co-written the article. JN has supervised the experiments, given expertise and co-written the article.

FUNDING

This work has been funded by the University of Uppsala.

[19] http://www.amateur-writing.com/

[20] http://acriticismlab.org/

REFERENCES

Chowdhury, G. (1999). The internet and information retrieval research: a brief review. *J. Document.* 55, 209–225. doi: 10.1108/EUM0000000007144

Clarke, S. J., and Willett, P. (1997). Estimating the recall performance of Web search engines. *Proc. Aslib* 49, 184–189. doi: 10.1108/eb051463

Croft, B., Metzler, D., and Strohman, T. (2010). *Search Engines: Information Retrieval in Practice: International Edition*, Vol. 54 (Boston, MA: Pearson Education).

Dubremetz, M. (2013). "Vers une identification automatique du chiasme de mots," in *Actes de la 15e Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'2013)*, eds. E. Morin and Y. Estève (New York, NY: Association pour le Traitement Automatique des Langues), 150–163.

Dubremetz, M. (2018). *mardub1635/corpus-rhetoric: First Release*, (Uppsala).

Dubremetz, M., and Nivre, J. (2015). "Rhetorical figure detection: the case of chiasmus," in *Proceedings of the Fourth Workshop on Computational Linguistics for Literature* (Denver, CO: Association for Computational Linguistics), 23–31.

Dubremetz, M., and Nivre, J. (2016). "Syntax matters for rhetorical structure: the case of chiasmus," in *Proceedings of the Fifth Workshop on Computational Linguistics for Literature* (San Diego, CA: Association for Computational Linguistics), 47–53.

Dubremetz, M., and Nivre, J. (2017). "Machine learning for rhetorical figure detection: more chiasmus with less annotation," in *Proceedings of the 21st Nordic Conference of Computational Linguistics* (Gothenburg: Linköping University Electronic Press), 37–45.

Dunn, J. (2013). "What metaphor identification systems can tell us about metaphor-in-language," in *Proceedings of the First Workshop on Metaphor in NLP* (Atlanta, GA: Association for Computational Linguistics), 1–10.

Gawryjolek, J. J. (2009). *Automated Annotation and Visualization of Rhetorical Figures*. Master thesis, Universty of Waterloo.

Hammond, A., Brooke, J., and Hirst, G. (2013). "A tale of two cultures: bringing literary analysis and computational linguistics together," in *Proceedings of the Workshop on Computational Linguistics for Literature* (Atlanta, GA: Association for Computational Linguistics), 1–8.

Hromada, D. D. (2011). "Initial experiments with multilingual extraction of rhetoric figures by means of PERL-compatible regular expressions," in *Proceedings of the Second Student Research Workshop associated with RANLP 2011* (Hissar), 85–90.

Kelly, A. R., Abbott, N. A., Harris, R. A., DiMarco, C., and Cheriton, D. R. (2010). "Toward an ontology of rhetorical figures," in *Proceedings of the 28th ACM International Conference on Design of Communication, SIGDOC '10* (New York, NY: ACM), 123–130.

Koehn, P. (2005). "Europarl: a parallel corpus for statistical machine translation," in *The Tenth Machine Translation Summit* (Phuket), 79–86.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). "The stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (Baltimore, MD: Association for Computational Linguistics), 55–60. doi: 10.3115/v1/P14-5010

Neufeldt, V., and Guralnik, D. B. (1997). *Webster's New World College Dictionary*. Indianapolis, IN; Chichester: Webster's New World; Macmillan.

Pasanek, B., and Sculley, D. (2008). Mining millions of metaphors. *Lit. Linguist. Comput.*, Vol. 23, 1–32. doi: 10.1093/llc/fqn010

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830. Available online at: https://hal.inria.fr/hal-00650905v1

Strommer, C. W. (2011). *Using Rhetorical Figures and Shallow Attributes as a Metric of Intent in Text*. Ph.D. thesis, University of Waterloo.

Tan, L. (2015). *A Corpus of Quotes*. Available online at: https://github.com/alvations/Quotables

Vandendorpe, C. (1991). Lecture et quete de sens. *Protée* 19, 95–101.

Zhang, E., and Zhang, Y. (2009). "Average precision," in *Encyclopedia of Database Systems*, eds L. Liu and M. T. Özsu (Boston, MA: Springer), 192–193.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.