Fixing Comma Splices in Italian with BERT

Daniele Puccinelli
DTI-ISIN
University of
Applied Sciences
of Southern Switzerland
Manno, Switzerland

Silvia Demartini
DFA-DILS
University of
Applied Sciences
of Southern Switzerland
Locarno, Switzerland

Renée E. D'Aoust
North Idaho College
Sandpoint, Idaho, USA
Casper College
Casper, Wyoming, USA

Abstract

We propose a fully unsupervised strategy to fix comma splices. Leveraging the pre-training of Bidirectional Encoder Representations from Transformers (BERT), our strategy is to mask out commas and let BERT guess what to replace them with. Our strategy achieves promising results on a challenging targeted corpus of awkwardly worded sentences from Italian-language college student essays.

1 Introduction

Comma splices can be defined as independent clauses joined by a comma without a coordinating conjunction (Hacker, 2009). Comma splices are frequent in both English and Italian and typically suggest a lack of basic understanding of sentence structure. As we will show, they come in various flavors, and there exist subtle differences between how they occur in English and Italian.

Comma splices are generally detected by commercial grammar and style checkers, but their automated correction has only been addressed by a few studies specific to English. Because the common denominator shared by such studies is the use of supervised machine learning techniques, the key research question that motivated the present study is whether we can use transfer learning to correct comma splices automatically in a completely unsupervised fashion and in languages other than English.

Thanks to contextualized word embeddings, and, in particular, thanks to BERT (Devlin et al., 2019), we show that it is possible to correct common cases of comma splices in Italian. We also discuss the limitations of our unsupervised approach.

2 Comma splices in Italian

Comma splices are widespread in contemporary written Italian language usage due to a tendency to over-extend the use of commas (Ferrari, 2017, 2018; Demartini and Ferrari, 2018). Several authors have studied this tendency in recent years. Some preserve the English language designation; this is the case in (Corno, 2019), where the expression *frasi fuse* (fused sentences) is also employed. Others employ alternate designations, such as *virgola passe-partout* (passe-partout comma) in (Tonani, 2010) and *virgola tuttofare* (factotum comma) in (Serianni and Benedetti, 2009).

Comma splices are one of the most frequent comma usage errors in Italian, especially among inexperienced L1 and L2 writers. Comma splices are also one of the principal and most common problems in the writing of university students, especially in science and engineering. Usually, these writers have failed to develop any linguistic awareness for text segmentation and organization, and they mistakenly assume that a comma can convey multiple functions, working both as a linker or as a strong stop.

There are some similarities and some differences compared to English usage, due to the fact that Italian punctuation is more communicative and less morphosyntactic. In gen-

eral, there are two main kinds of comma splices in Italian that are caused by the use of a comma where we would expect:

1. a logical connector to join two sentences that have a particular relationship;

2. a stronger punctuation mark to mark a logical-syntactic connection (colon) or break (semicolon or period).

According to (Ferrari, 2014), comma splices reflect a deep inability to handle both basic syntactic structures and text construction: if a text is characterized by coherence, cohesion, and topical organization, comma splices deconstruct these properties from the inside. For this reason, analyzing comma splices is extremely important in the context of improving language teaching.

Comma splices can be fixed in various ways, depending on the context and on the kinds of clauses involved. In the most straightforward cases, the comma can be replaced by a period or a semi-colon that explicitly separates the clauses on either side of the comma. In other cases, the comma can be replaced by an element that links the clauses, such as a colon, a conjunction, or a conjunctive adverb. Care must be exercised if sentences are more complex (i.e. with parenthetical elements) or syntactically inaccurate.

Due to the lack of an Italian-language corpus dedicated to comma splices, the authors have assembled a small corpus of 100 sentences containing a wide array of comma splices collected from college student writings (mostly in the field of engineering) at the Università del Piemonte Orientale (UPO) and the University of Applied Sciences of Southern Switzerland (SUPSI) in the mid-to-late 2010s. In the remainder of the paper, we will employ this UPO-SUPSI-SPLICE corpus (henceforth USS corpus) to evaluate the potential of our proposed method. Aside from containing comma splices, many USS sentences are poorly worded, syntactically inaccurate, and often unclear.

3 Related work

In the active research thread on automated grammar and style correction, the studies that are most closely related to ours are (Lee et al., 2014) on the automated detection of comma splices and, most recently, (Zheng et al., 2018) on the automated correction of run-on sentences. The techniques proposed in these studies, which are specific to English, rely on supervised learning techniques that require relatively extensive training sets. To the best of our knowledge, ours is the first investigation of the automated correction of Italian-language comma splices using unsupervised learning.

Our proposed unsupervised strategy leverages the rich research thread on word embeddings. Dense word embeddings went mainstream with Word2Vec (Mikolov et al., 2013) and gained traction in the mid-to-late 2010s in spite of their key limitation that a word type has the same word embedding regardless of context. Because words also have different aspects depending on semantics, syntactic behavior, and register/connotations, contextualized word embeddings have emerged as an elegant solution to capture word semantics across different contexts. TagLM (Peters et al., 2017) uses the hidden state of the bidirectional long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) as a contextual word embedding. Instead of just using the output of the LSTM, ELMo (Peters et al., 2018) uses all the available hidden layers and combines them in a task-specific way with task-dependent trainable weights that can be learned for each task. ELMo embeddings have been shown to improve the state-of-the-art on a wide variety of challenging NLP tasks, but even more significant improvements have been shown with BERT (Devlin et al., 2019). Based on Transformer encoders (Vaswani et al., 2017), which are essentially a multi-headed attention stack where depth serves to compensate for the lack of recurrence, BERT pre-trains bidirectional representations by jointly conditioning on both the left and right context of individual tokens, and

allows for low-cost task-specific fine-tuning.

4 Fixing Comma Splices with BERT

While bidirectionality comes naturally to LSTM-based models, it is challenging to achieve it with Transformer-based models, because bidirectional conditioning with multiple layers inherently allows each word to see itself. BERT's solution is to mask a relatively small portion of the tokens in the pre-training data and to train a bidirectional language model to guess them. If too few words are masked, training is too expensive, while if too many words are masked, BERT fails to learn about language; it was determined empirically that masking 15% of all tokens represents a reasonable compromise.

This specific aspect of BERT's pre-training means that a pre-trained BERT model has the ability to predict missing tokens out of the box, i.e., with no task-specific fine-tuning and, therefore, no need for task-specific training data. For our purposes, this translates into a straightforward strategy to correct comma splices: **mask all commas** and **use BERT to guess what they should be**. In principle, if a masked comma is legitimate, we expect BERT to guess it is indeed a comma, while if it is not, as in a comma splice, we expect BERT to replace it with a more appropriate token.

BERT naturally lends itself to this task because it outputs an empirical probability distribution over a set of potential replacement tokens. Such tokens can be drawn out of the entire dictionary (including word pieces) or over a controlled subset. Jointly with the probabilistic nature of its output, BERT's inherent bidirectionality may be directly harnessed by making predictions based on both the left and the right context of a masked comma and choosing the set of predictions associated with the highest probability.

If the array of potential replacement token is unrestricted, in complex sentences BERT may elect to replace commas with tokens belonging to inappropriate word classes, such as nouns or verbs. This can be avoided by restricting the eligible potential replacement tokens to reasonable word classes.

Strategy	Accuracy
Baseline	0.41
BERT - left context only	0.77
BERT - left & right context	0.81
BERT - PoS + left & right	0.87

Table 1: Sentence-level accuracy for the baseline strategy and the three different flavors of our BERT-based strategy described in this paper, measured on the USS corpus.

5 Evaluation

As a proof of concept, we perform an empirical evaluation of our BERT-based strategy on the USS corpus, which contains sentences with at least one comma splice and a total number of commas ranging from one to seven. To the best of our knowledge, no directly comparable technique to fix Italian-language comma splices programmatically is freely available at the time of writing. To get a rough idea of the potential of our strategy, we use a simple baseline that replaces all commas with periods. While this baseline fails each time a sentence contains multiple commas, it fixes over 90% of the USS sentences that contain exactly one comma (41 out of 45). Aside from setting a performance floor, this baseline also offers a quick idea of the complexity of the sentences in the corpus.

As for our BERT-based strategy to fix comma splices, we make the following choices for the sake of simplicity:

- we employ `bert-multilingual-uncased` (and normalize all tokens to lower case);
- we draw potential replacement tokens out of the entire dictionary (aside from the PoS-based restrictions described below), but only consider potential replacement tokens with an estimated probability greater than 0.01 (arbitrary threshold);
- we make predictions based on both the

left and the right context of the masked tokens and choose the prediction associated with the highest probability, computed as the product of the probabilities of the most probable token replacement for each comma occurrence (we always mask out one comma at a time);

- we use PoS tags to exclude potential replacement tokens from word classes other than conjunctions and punctuation marks.

We employ TreeTagger to determine the PoS tags and use pre-trained BERT by way of `pytorch_pretrained_bert`.

We use sentence-level accuracy as our figure of merit and compute it as the fraction of error-free corrected sentences. A sentence is considered to be error-free by our strategy and/or by the baseline if the corrected version is acceptable according to two L1 human annotators. The corrected versions of sentences with multiple commas are only considered error-free if they contain no anomalies; while this is overly penalizing for our strategy in multi-comma sentences where a single mistake is made, it offers a conservative estimate of the performance of our BERT-based strategy.

As shown in Table 1, our BERT-based strategy is able to correct a total of 87 of the 100 sentences in the USS corpus to the satisfaction of the two L1 human annotators. An additional sentence is also corrected, but only if our strategy operates unidirectionally.

6 Discussion

Commas per sentence. The mean number of commas is 2.1 in the sentences where our strategy succeeds, while it is as high as 3.5 in the 12 sentences where our strategy fails. While multi-comma sentences are inherently more challenging, there doesn't seem to be a hard limit to the number of commas per sentence that our strategy can handle. Notably, our corpus contains a 7-comma excerpt:

> *Di solito, chi scrive senza conoscere le fasi della scrittura, scrive di getto, seguendo i propri ragionamenti senza un ordine, così facendo, rischia di non scrivere un testo idoneo e fluente, dobbiamo essere attenti alle punteggiature, non scrivere le frasi molto lunghe e dividere in modo adeguato i capoversi.*

which is fixed as

> *Di solito, chi scrive … scrittura, scrive di getto, seguendo … **ordine.** **Così** facendo, rischia … **fluente.** **Dobbiamo** … punteggiature, non … capoversi.*

Failures in single-comma sentences. There are two single-comma sentences where our strategy fails: one contains a run-on sentence and also causes the baseline strategy to fail, while the other one has a mild form of comma splice:

> *Successivamente avviene la documentazione, si raccolgono e si scelgono le informazioni da fonti attendibili e si pianifica come esporle.*

This sentence is the only instance in USS where our strategy fails and the baseline strategy succeeds. BERT chooses not to replace the comma, keeping the (borderline acceptable) comma splice unaltered. This happens due to the relative values of the probabilities assigned by BERT to a comma and a colon. Curiously, replacing *Successivamente* with the equivalent expression *Al passo successivo* is enough to nudge BERT in the right direction and assign a higher probability to a colon. This suggests that modifying individual tokens in a small corpus such as USS would be a meaningful dataset augmentation technique.

Left and right context. For 77 out of 100 sentences, a unidirectional pass based on the left context of the missing tokens is sufficient for our strategy to succeed. Only one of these 77 sentences can only be corrected unidirectionally; five other sentences can also be corrected by looking at the right context of the

missing tokens in a backward pass, which helps avoid blatantly erroneous replacements. Therefore, our strategy should be used with both left and right context. As an example, consider the sentence:

> *Essa consiste nel fatto che non c'è alcun legame naturalmente motivato, il significante cane non ha di per sé nulla che rimandi al suo nome, che faccia sì che quella cosa si possa chiamare così.*

If BERT only relies on the left context of missing commas, the sentence is awkwardly split into three parts, with a striking error at the end:

> *Essa . . . motivato.* ***Il*** *significante . . . al suo nome.* ***Che*** *faccia sì che quella cosa si possa chiamare così.*

With both left and right context, instead, our strategy offers an acceptable correction:

> *Essa . . . motivato.* ***Il*** *significante . . . al suo nome* ***e che*** *faccia . . . così.*

PoS filtering. A further six USS sentences can be fixed by combining left & right context and PoS filtering, which serves to avoid replacement tokens from implausible word classes and prevent awkward errors, such as the replacement of a comma with a preposition, a *che*, or a negation. Comma replacements with negations are particularly critical because they modify the meaning of the corrected sentence. PoS filtering is also helpful to prevent BERT from replacing commas with word pieces, which may occur with awkwardly worded sentences.

Unacceptable replacements. We have observed a limited number of unacceptable replacements of commas with colons, all of which occur in long-winded multi-comma sentences. Consider the six comma sentence:

> *Per la creazione della piattaforma web, il committente ha desiderato utilizzare una web application in Java, avendo la possibilità di scegliere tra due framework, Spring e Struts, si è optato per l'utilizzo di Spring, siccome è uno strumento già utilizzato precedentemente, possiede un'ottima documentazione.*

which becomes:

> *Per la creazione della piattaforma* ***web.*** ***Il*** *committente . . . in* ***Java,*** *avendo la possibilità di scegliere tra due* ***framework:*** *Spring e Struts.* ***Si*** *è optato per l'utilizzo di* ***Spring,*** *siccome è uno strumento già utilizzato precedentemente* ***e*** *possiede . . .*

The first comma is erroneously replaced with a period, and the third one is questionably replaced with a colon. Replacing the fourth comma with a period is acceptable, as is preserving the second and fifth commas and turning the sixth comma into an *e*. Though our strategy makes four correct decisions out of six, this example is considered incorrect for our sentence-level quantitative analysis.

Interestingly, we have not observed any replacements with semicolons. We conjecture that BERT's strong preference for colons may be due to the relative frequency of colons versus semicolons in the pre-training text.

7 Conclusion

While the main limitation of the present study is the limited size of the USS corpus, we believe that the challenging nature of the writing excerpts in the USS corpus has enabled us to stress-test our strategy and to deliver a solid proof of concept that leverages the power of BERT-style contextualized word embeddings for automated style correction. Our future plans include using our BERT-based strategy to correct comma splices in English-language L1 and L2 student writing and to correct run-on sentences.

References

D. Corno. 2019. *Scrivere e comunicare. La scrittura italiana in teoria e in pratica*. Pearson.

S. Demartini and P. Ferrari. 2018. La virgola splice nei testi di studenti universitari: un problema solo in apparenza superficiale. In *La Punteggiatura Italiana Contemporanea nella Varietà dei Testi Comunicativi*.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL '19*.

A. Ferrari. 2014. *Linguistica del testo. Principi, fenomeni, strutture.*. Carocci.

A. Ferrari. 2017. Usi "estesi" del punto e della virgola nella scrittura italiana contemporanea. *La lingua italiana. Storia, strutture, testi* XIII:137–153.

A. Ferrari. 2018. *La punteggiatura italiana contemporanea. Unanalisi comunicativo-testuale*. Carocci.

D. Hacker. 2009. *The Bedford Handbook for Writers*. Bedford Books.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8).

J. Lee, C. Y. Yeung, and M. Chodorow. 2014. Automatic detection of comma splices. In *28th Pacific Asia Conference on Language, Information and Computation pages (PACLIC '14)*.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS '13*.

M. Peters, W. Ammar, C. Bhagavatula, and R. Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *ACL'17*. Vancouver, Canada.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL'18*. New Orleans, Louisiana.

L. Serianni and G. Benedetti. 2009. *Scritti sui banchi. L'italiano a scuola fra alunni e insegnanti*. Carocci.

E. Tonani. 2010. *Il romanzo in bianco e nero. Ricerche sull'uso degli spazi bianchi e dell'interpunzione nella narrativa italiana dall'Ottocento a oggi*. Cesati.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *NIPS '17*.

J. Zheng, C. Napoles, J. Tetreault, and K. Omelianchuk. 2018. How do you correct run-on sentences it's not as easy as it seems. In *The 4th Workshop on Noisy User-generated Text W-NUT*. Brussels, Belgium.