**Mark Boardman**
**m.boardman@hud.ac.uk**
**markboardman@outlook.com**
**+44 (0)7780 515377**
**PhD candidate in the Linguistics And Modern Languages Department at The University Of Huddersfield**

**Talk delivered at PALA 2019, University of Liverpool, UK**

**Title**
Grammatical agency and ironic persona in Emily Dickinson: an interdisciplinary corpus originated study.

**Abstract**
This study uses a combination of computational, corpus and qualitative stylistic methods to isolate and analyse, from my own corpus of Emily Dickinson's 'fascicle' poems (Miller, 2016), some syntactic and morphological markers which potentially construct the effect of an ironic authorial persona. Based on the results of a user-designed set of syntactic and morphological queries performed on the corpus by a rule based NLP software package called NooJ (Silberztein, 2016), I intended to suggest ways in which selected concepts from Cognitive Grammar (Langacker, 2009) might be used to develop those computational results into a qualitative theory of how grammatical markers related to person and agency can be seen to instantiate some forms of irony. That it may be possible for corpus tools to point towards a link between syntax and irony is suggested by Luow (in Baker et al, 1993), but most of the software tools and packages commonly used in corpus based literary stylistics focus on lexical and semantic variables and do not incorporate features capable of analysing complex syntactic patterns. As NooJ is not reliant on stochastic algorithms and is entirely rule based, it allows the user to design bespoke syntactic and morphological queries as complex as the user's knowledge of grammar will facilitate. Because these queries are user-designed, erroneous results can be filtered out systematically by debugging the rules in the same way as a programmer would debug computer code.

**Keywords**
irony; persona; corpus; literary; stylistic; computational; NLP; NooJ; syntax; morphology; cognitive

**Irony and qualitative analysis**
Irony is not an extensively explored topic within stylistics, and as far as I am aware has only been explored within the sub-discipline of corpus stylistics in Bill Luow's 1993 paper "Irony in the Text or Insincerity in the Writer? The Diagnostic Potential of Semantic Prosodies". My particular interest is in investigating whether irony could be shown to be instantiated syntactically as well as lexically and semantically, and Luow hints fleetingly that this might be possible:

> 'As the research stands at present, it looks as though the prosodies based on very frequent forms can bifurcate into 'good' and 'bad', using a grammatical principle like transitivity in order to do so. For example, where *build up* is used transitively, with a human subject, the form of the prosody is uniformly good. People *build up organisations, better understanding* and so on. Where things or forces, such as *cholesterol, toxins,* and *armaments build up* intransitively, of their own accord, they are uniformly bad.' (Luow, 1993)

This is a broadly analogous linguistic principle to my intuition that a first person persona adopting a stance perceived to be ironic in Dickinson's work frequently uses lexical ambiguity in conjunction with reversal of grammatical agency. As McIntyre and Walker (2019) comment '...what makes corpus analytical techniques particularly valuable in stylistics is that they can be used to: identify patterns in texts; determine norms; validate or invalidate intuitions...'. I have so far successfully set up the prototypical computational mechanisms for identifying the patterns, and the final two years of my part time PhD programme will be spent determining the norms and investigating the validity of the intuitions.

It is worth considering some approaches to irony from qualitative stylistics, one of the best known being that put forward by Paul Simpson in his 2011 paper "'That's not ironic, that's just stupid': Towards an eclectic account of the discourse of irony':

> '...there is a sense in much linguistics research that irony is a pragmatic device which in terms of its transmission and retrieval is essentially binary in nature; that is, a form of figurative language which splits

> easily into that which is ironic and that which is not. This paper has striven to reject binarism in favour of a more pluralist approach which favours construction and consensus in the development of irony; an approach where speech acts interact with context and situation often in very complex and finely-shaded ways.' (Simpson, 2011)

Stylistic analysis that is purely qualitative and not informed by corpus based research, is very capable of reflecting fine shading in its manipulation of analytical concepts, but there are potential problems when trying to reconcile quantitative results from corpus analysis with a qualitative framework. Simpson argues persuasively for a multi-faceted, socially motivated definition of irony. A possible objection to this non-binary approach, though, is that it is not easily replicable or falsifiable:

> '…I was unsure that irony was objectively separable from some other incongruities in communication and there are still individual cases which make it difficult to see where the boundaries are, at times, between irony and other forms of linguistic or situational clash, such as hypocrisy, paradox, punning or ambiguity.' (Jeffries, 2018)

Most agree that some form of clash is taking place between two opposing propositions, but a further issue suggested indirectly by Simpson and Jeffries is that perhaps the term 'irony' is not very useful critically, bound up as it is with social and situational assumptions and preconceptions which are difficult to integrate into formal linguistic analysis. McIntyre and Walker appear to rest with an essentially binary approach:

> '….it is a matter of comparing the semantic prosody of a particular unit of meaning against an implicature arising from a specific phrase and discovering that the propositional assertions of the two positions are incongruous. This is broadly in line with Simpson's assertion that verbal irony is 'a perceived conceptual space between what is asserted and what is meant'.' (McIntyre & Walker, 2019:61)

The problem for a qualitative analyst is how to characterise such incongruity or conceptual space in ways that another critic can replicate and apply rigorously to any text, and the problem for a corpus analysist is how to frame a computational query that will reliably identify all relevant instances of such incongruity or conceptual space in a corpus. This points to the more fundamental methodological conundrum of how statistical results from corpus analysis can be reliably connected to a qualitative framework capable of developing those results into productive observations on reader response.

It had been my intention to devise a way of linking the quantitative results from my corpus analyses to Langacker's (2009) Cognitive Grammar framework, for the simple reason that I have always found his concept of *construal* to be a very persuasive device for elucidating reader response qualitatively, and so wanted to use it in conjunction with my more recent interest in corpus and computational stylistics. However, I have been unable to surmount the fact that Cognitive Grammar is primarily metaphorical in its formulation and is therefore not easily adapted to the ways in which NLP and corpus processing software carry out their analysis – grounded as they still are in Chomsky's (1957, 1965) early work on formal and generative grammatical concepts.

Despite some attempts to formulate cognitive linguistic models computationally, most applications of NLP are still based on the premise that a given language stream (written or spoken) is a manifestation of surface form whose meanings and structures can be accounted for by referring to rules of well-formedness. With this in mind, the computer needs a *grammar* with which to interpret a language stream:

> 'When we refer to formal language, it is intuitive to evoke the concept of formal grammar. In fact, a formal language can be seen as the product of a grammar that describes it. Formally, such a grammar is defined by a quadruplet: $G = (V_N, V_T, P, S)$ where: – $V_N$: the non-terminal vocabulary; – $V_T$: this vocabulary brings together all of the terminals of the grammar, which are commonly called the words of the language; – P: the set of the rewrite rules of grammar (production rules); – S: sometimes called an axiom, it is a special element of the set $V_N$ which corresponds to well-formed sentences.' (Kurdi 2016: 153)

In essence, *terminals* are words and *non-terminals* are the codes that the grammar applies to the language stream in order to decide whether to output segments within that stream as terminals. For example, if a NooJ grammar contains the non-terminal <ADJ> and when carrying out a user defined query it encounters the segment 'slow' in the language stream, by searching through the relevant dictionary file for that word and referencing it against a specified morphological definitions file, it will decide to output 'slow' as a terminal in the resulting concordance.

Given that this is the way NLP software works, including tools used by stylisticians for analysing corpora, it seems logical to devise a qualitative analytical method based on the way a computer implements formal grammar. This remains an undeveloped idea currently, but the principles I will need to discuss qualitatively, once the quantitative analysis of the corpora has been completed, are as follows: to what extent does the data show conflicting attitudes to the self, and to what extent are conflicting attitudes to the self instantiated through a

combination of semantic and syntactic mechanisms? As part of the process of formulating a usable qualitative method, I will need to decide whether to try to reference my analyses against established stylistic definitions of irony, or indeed whether to use the word 'irony' at all.

**Methodology and prototypical data analysis**

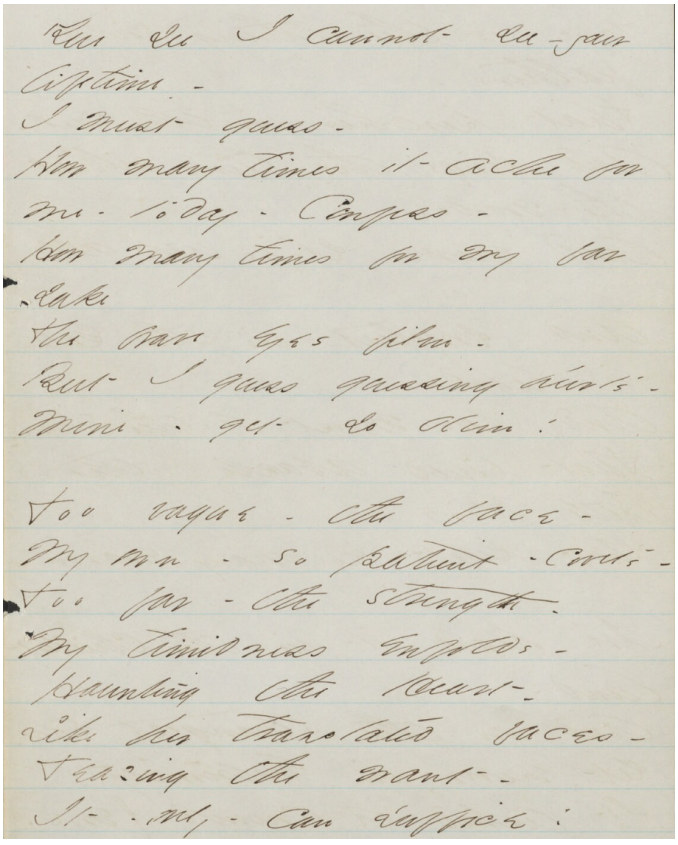
The principle of *semantic oscillation*, yet to be developed, will be defined as the simultaneous co-existence of two or more meanings in one lexical item, as with the word 'kindly' in the following couplet from fascicles poem 443:

'Because I could not stop for Death —
He kindly stopped for me —'

*Agency reversal* will be defined using a definition of 'agent' as the person or entity performing the action of the verb. In the example above, the first person pronoun begins as the subject and agent of a modal state, and abandons both agency and the subject position in the sentence to a third party, Death. It is these two principles that I aim to account for computationally, using NooJ, and qualitatively, using a framework yet to be determined.

In order to reliably identify agency reversal, it is first necessary to develop a query for identifying where first person agency occurs in subject positions in the corpus. To test this principle, I created a prototypical micro-corpus from fascicles poem 0273 and designed a NooJ query to elicit a concordance of all the instances, in the micro-corpus, of first person agents in the subject position.

The first process is electronic tokenisation of the poem from its handwritten manuscript:



**You see I cannot see — your lifetime —**
**I must guess —**
**How many times it ache for me — today — Confess —**
**How many times for my far sake**
**The brave eyes film —**
**But I guess guessing hurts —**
**Mine — get so dim!**

**Too vague — the face —**
**My own — so patient — covets —**
**Too far — the strength —**
**My timidness enfolds —**
**Haunting the Heart —**
**Like her translated faces —**
**Teasing the want —**
**It — only — can suffice!**

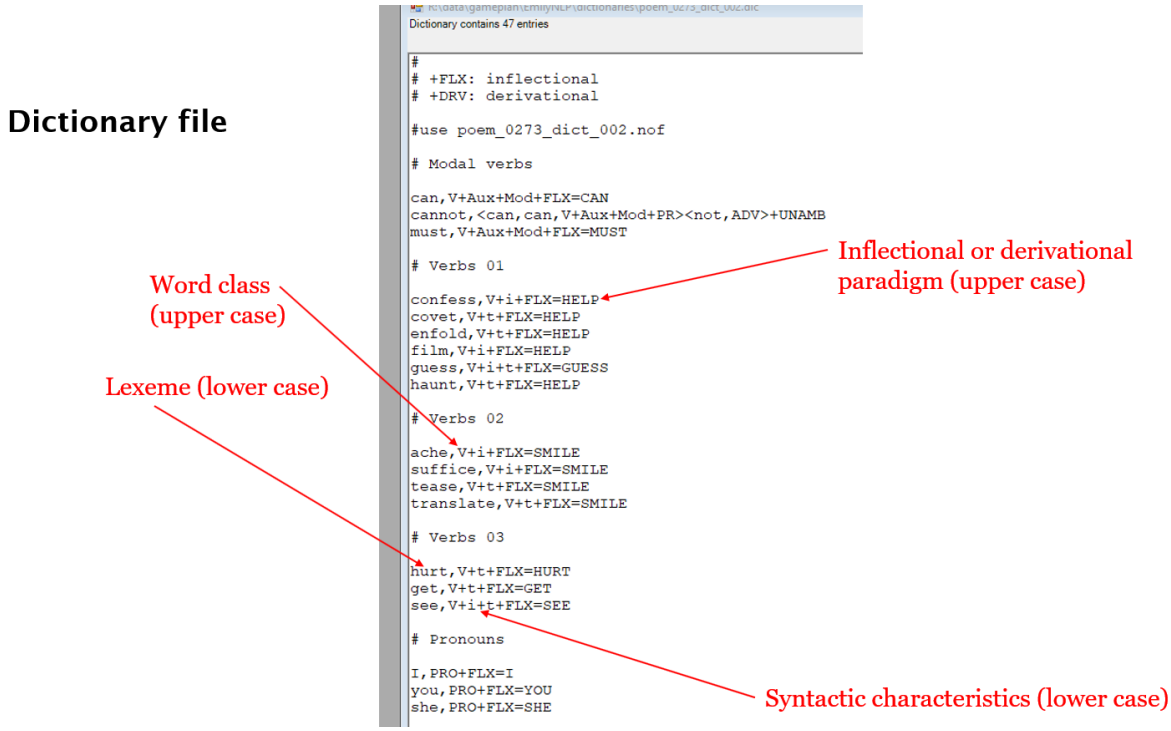
NooJ will import any text file format, without the need to convert to plain text or add any special characters. It will also recognise XML tags and treat them as structurally meaningful, so I marked up the micro-corpus in the following way:

**<poem>**
**<stanza>**
**<line>You see I cannot see — your lifetime —</line>**
**<line>I must guess —</line>**

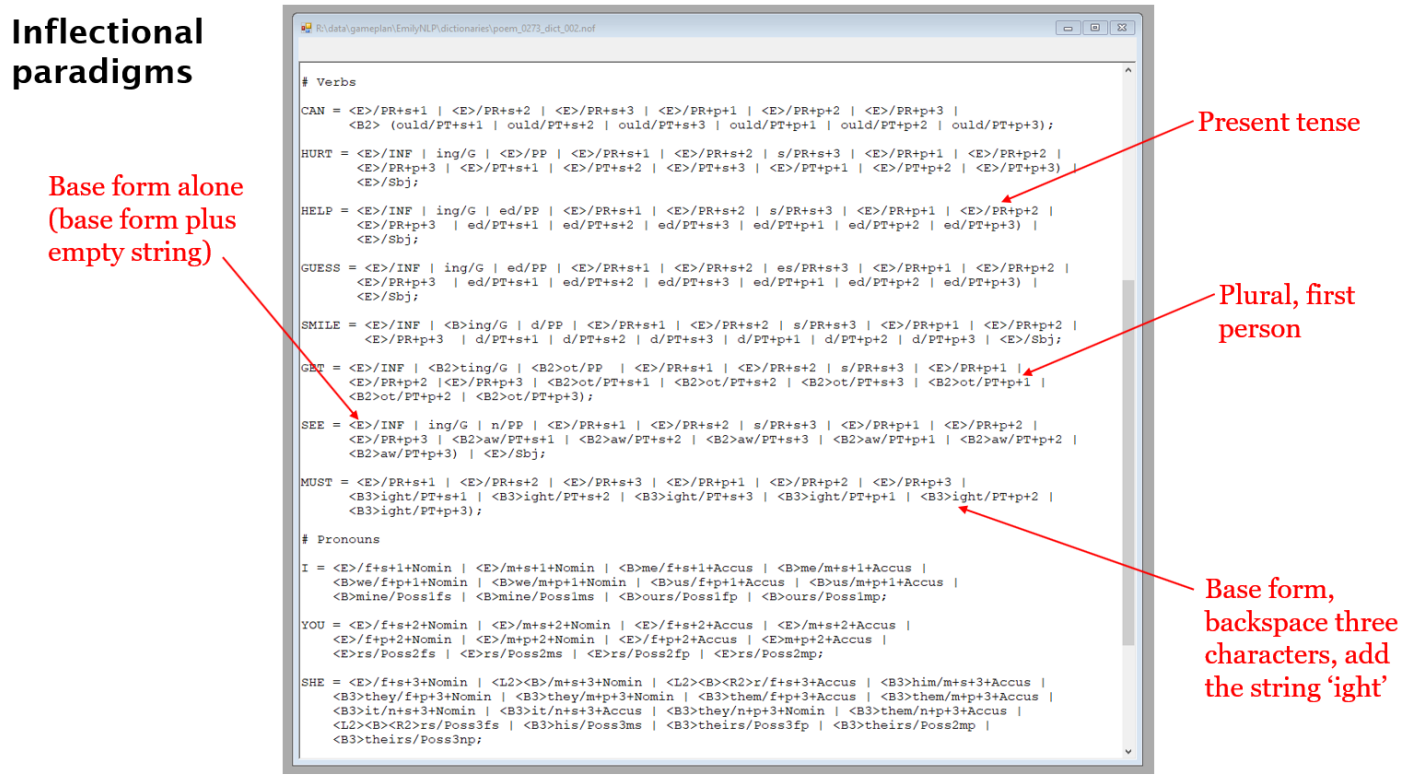
<line>How many times it ache for me — today — Confess —</line>
<line>How many times for my far sake</line>
<line>The brave eyes film —</line>
<line>But I guess guessing hurts —</line>
<line>Mine — get so dim!</line>
</stanza>
<stanza>
<line>Too vague — the face —</line>
<line>My own — so patient — covets —</line>
<line>Too far — the strength —</line>
<line>My timidness enfolds —</line>
<line>Haunting the Heart —</line>
<line>Like her translated faces —</line>
<line>Teasing the want —</line>
<line>It — only — can suffice!</line>
</stanza>
</poem>

The full versions of the five sub-corpora are marked up in this way throughout, in order to facilitate the future development of queries that might need to isolate individual poems, stanzas or lines.

NooJ needs to refer to a dictionary file in order to recognise valid lexical items within the imported text. This is an extract from the prototype dictionary file I developed using all of the unique lexical items in fascicles poem 0273:

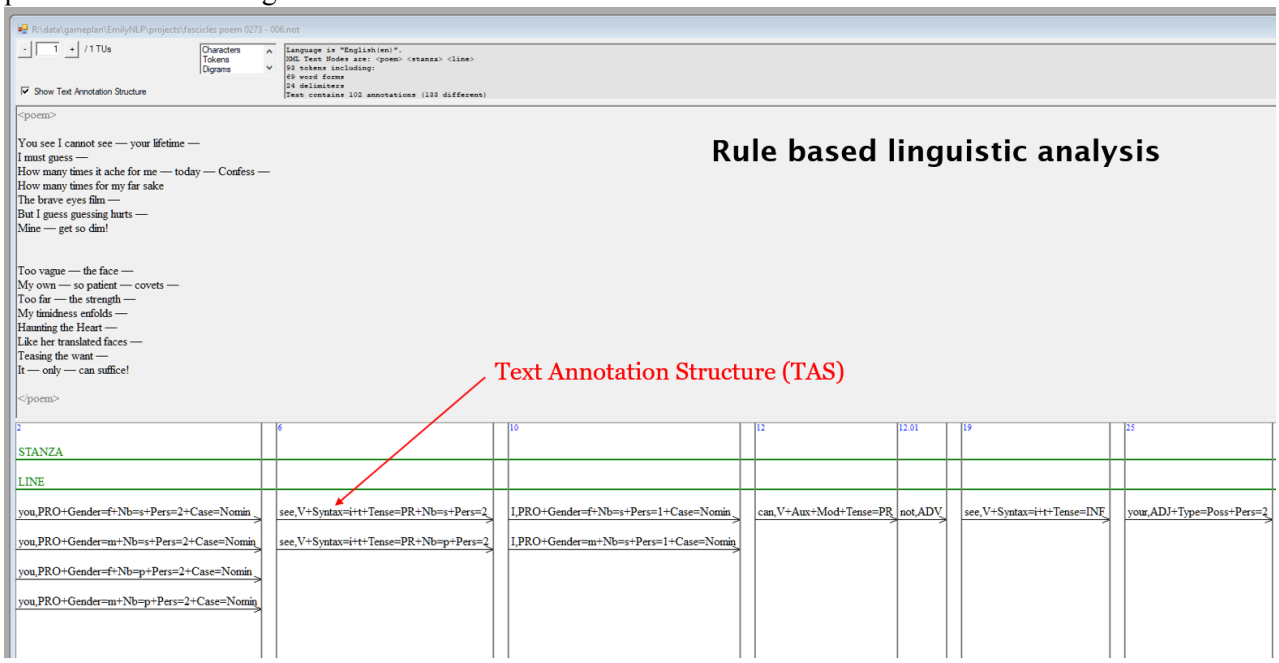



Lexical items are listed in lower case, but do not need to be in alphabetical order. For each entry, after the comma the word class that NooJ should attach to occurrences of that word is indicated. This can be important as part of disambiguation, given that for example ‘haunt’ can either be a noun or a verb. Indicated after plus signs, in lower case, are any syntactic characteristics that NooJ is required to follow for a given lexical item – in the highlighted example, clarifying that ‘see’ can be treated either transitively or intransitively. After capitalised ‘FLX’ followed by ‘=’ is an instruction as to which inflectional pattern the word should follow in the morphological definitions file ‘poem_0273_dict_002.nof’ indicated at the top of the dictionary file:




Listed down the left side are the base forms of the paradigms referred to for each entry after 'FLX=' in the dictionary file. '<E>' indicates, for the paradigm 'SEE', that the base form followed by nothing (an empty string) can be treated as, for example, the infinitive form, the present first person singular form, or as the present second person singular form. On the second line of the 'SEE' entry, <B2> instructs NooJ to backspace two characters, deleting 'EE' to leave 'S' which followed by 'aw' forms the past tense first person singular form.

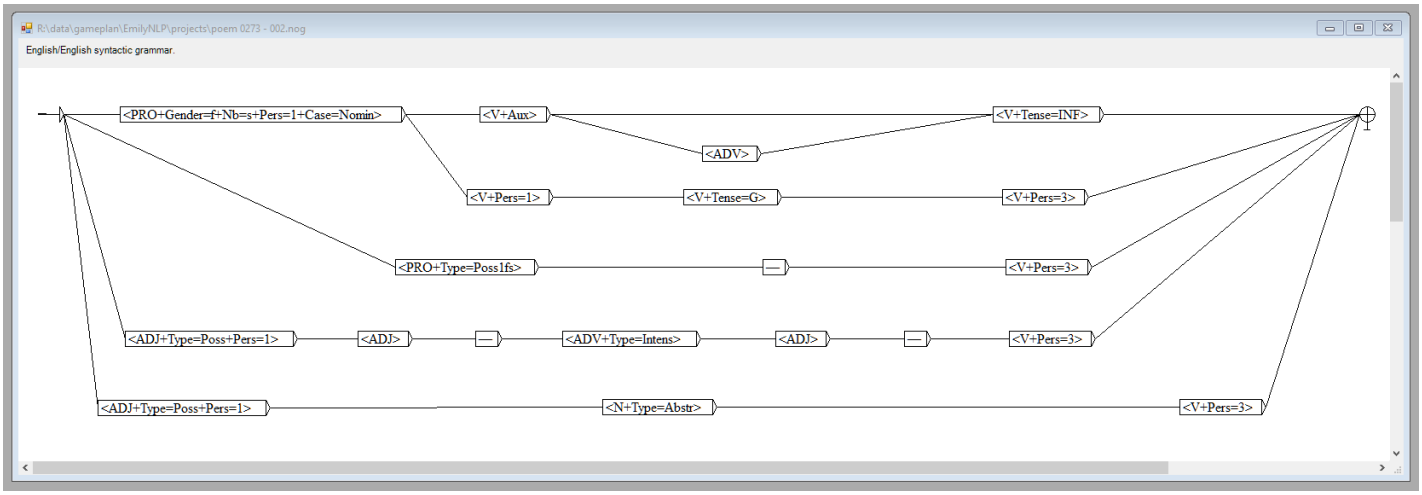
Once NooJ has been instructed to use the dictionary file 'poem_0273_dict_002.dic' in its analysis, it produces the following text annotation structure:




Each segment in the surface form stream is assigned a word class label in uppercase, followed by a list of the syntactic and morphological characteristics which have led NooJ to assign that word class. Where ambiguity

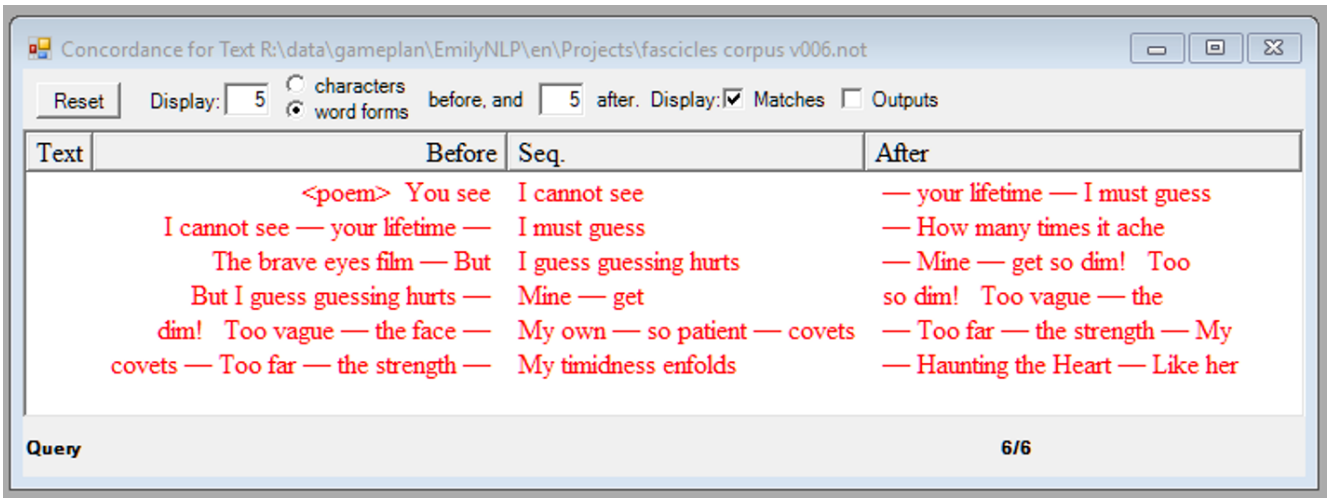
exists as to precisely how the word class label could be justified, all of the possibilities are included as alternatives.

The final and most meaningful step in designing the NooJ query to output a concordance of examples of first person grammatical agency, is to create a grammar file from a syntactic flow chart, using NooJ's graph editor:

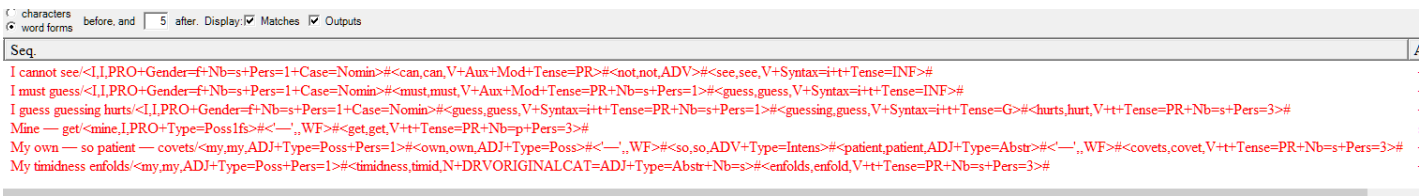



This grammar looks for SV combinations with first person forms (nominative pronoun forms, possessive pronouns and possessive adjectives) in the subject position. NooJ will output all sequences that match any of the left-to-right patterns specified in the flowchart, so it can be seen that the grammar file is a configurable template for analysing structure based on form (morphology) and distribution (syntax) according to principles outlined by Harris (1954) in the early days of structural linguistics.

The grammar produces the following concordance when applied to the micro-corpus developed from fascicles poem 0273:




If the 'outputs' box is ticked, the sequences in the concordance expand to include the morphological and syntactic basis on which NooJ has made the selections:




The next phase will be to develop and refine further grammar files to encompass agency reversal, and apply these grammar files to the whole corpus.

**The role of corpus processing**

NooJ is unique amongst software packages which are designed to allow the end user to query corpora linguistically, in being the only downloadable non-bespoke package available that is able to query corpora using complex morphological and syntactic criteria. NooJ is also almost completely unknown in the sub-field of corpus stylistics, largely because of its rule based approach. All of the widely used corpus processing tools are based on stochastic (statistical) principles which do not allow the analyst to query the text based on detailed considerations of word forms or sentence structure. Corpus tools are usually developed by computer scientists or linguists working in the field of Natural Language Processing, a field that has in recent decades moved away from rule based approaches. Topic modelling is a pursuit within NLP that aims to offer insights into text structure, but its core principles militate against fine grained linguistic analysis:

> 'Topic modelling works on a 'bag of words' principle. That is, it is linguistically naïve and pays no attention to the grammatical or semantic connections between words. Multiple estimation procedures have been proposed for topic models.' (Murakami et al, 2017)

This suggests that the underlying aim is to produce a coarse grained statistical analysis of occurrences and co-occurrences of topics in the dataset, rather than to look at sentence structure or (by implication) fine grained discourse structure. One grammatical feature that *is* available in commonly used corpus tools, is part-of-speech tagging. The user can load their corpus into the software environment, and the software will then attempt to tag each lexical item in that corpus with a label indicating that lexical item's word class, a process accomplished by comparing the user's inputted corpus to a large reference corpus and estimating the statistical likelihood of each word, in its context, belonging to a specific word class. If, though, we are looking for fine grained accuracy in setting up queries to analyse corpora, the essential problem is as follows:

> 'Taggers overlook the fundamental linguistic principle that sentences are structured objects, and that virtually any category of word can be inserted anywhere in a sentence without changing its structure. For example, after the verbal form sees, we might find an adjective (e.g. Joe sees red apples), an adverb (e.g. Joe sees very well), a determiner (e.g. Joe sees that apple), a coordinating conjunction (Joe sees but says nothing), a subordinating conjunction (Joe sees that Lea is happy), a noun (Joe sees flowers), a preposition (Joe sees behind himself), a pronoun (Joe sees that), a relative pronoun (Joe sees where Lea is going), or a verbal form (Joe sees destroyed buildings). In these conditions, disambiguating the word form sees (or any verbal form) on the basis of its contexts in a reference corpus – which necessarily possesses only a limited sample of all potential contexts – will produce a large number of errors.' (Silberztein, 2016)

For adherents of a rule based approach to NLP and corpus processing, the most reliable way to avoid large numbers of errors in corpus based analysis is to set up linguistic rules for the software to follow. Common objections raised to adopting this approach are: only expert linguists can set up meaningful rules for linguistic analysis; NLP as a sub-discipline has moved on beyond rule based analysis; setting up rules is labour intensive and time consuming; rule based software is not as fast as software driven by statistical algorithms and is therefore not as well suited to the analysis of distribution patterns in very large datasets. The last of these I fully accept: if the linguist's focus is only on large scale distributional features, a rule based algorithm will perform much more slowly. But in order to accomplish the fine grained syntactic and morphological analysis needed to investigate my research hypotheses, in relation to my modest corpus of 1800 poems, I need to spend the requisite time devising, setting up and debugging the rules.

**Next steps and interim conclusions**

Central to NooJ's functionality is the dictionary file, or lexicon. It is perfectly possible to do meaningful corpus analysis using the built-in English dictionary that comes with NooJ; but I consider that it would be stylistically more meaningful and precise to be working with only the lexical items that Dickinson employed, tagged, using NooJ conventions, according to the morphological rules evident in the corpus – given that a generic dictionary of English is very unlikely to fit precisely the usage habits and derivational possibilities with which every writer from every period would be familiar. There are just over 8000 unique lexical items in Dickinson's complete poems. These I have isolated as a list, and as the first phase of the tagging process I have run the list through TagAnt (Anthony, 2019), which incorporates TreeTagger (Schmid, 2019), a probabilistic part of speech engine. This has produced largely very accurate results, tagging each lexical item with TreeTagger labels which now need to be converted into codes that follow NooJ conventions. First the TagAnt results need to be manually error-checked, filtering out instances where, for example a nominal has been mis-identified as a verb and vice versa.

There is the question of whether to include a corpus of Dickinson's letters in the primary dataset, subject to time available. This I think would be potentially very productive, as it might reveal differences in how an ironic persona is constructed, if one is constructed at all, when Dickinson works within a medium that is much more directly interpersonal and transactional. Unlike the poems, which are available individually alongside individual manuscripts at The Emily Dickinson Archive, the complete letters are not available in any electronic form, so building a letters corpus is much more time consuming. I have produced scanned page images from the 1000 page Johnson (1958) edition of the letters, and if I proceed with building the letters corpus the next stage will be to run the image files though OCR software to extract the electronically tokenised text, followed by cleaning out the OCR errors and applying XML markup to the text.

Another key task will be to develop a qualitative analytical method to be applied to the results of the formal corpus based computational analysis. I am aware that many studies are only corpus based, and that statistical results are often essentially left to speak for themselves; but I retain a strong interest in elucidating reader responses to literary texts using linguistic tools, and still believe that stylistics can shed light on what literary critics notionally term an aesthetic response, so with that in mind I am committed to retaining the qualitative element of my research.

If time constraints allow, I have also considered potentially marking up my corpus in more complex ways and exploring how NooJ might be able to interact with this more complex markup. For example, I am interested in exploring the concept of *predicative triples*, first publicised in the context of Berners-Lee's (2001) work on the emergence of the 'semantic web'. Predicative triples are expressed in a development of XML called 'RDF' (Resource Description Framework) and are designed to describe relationships between people and actions in order to make web searching more accurate and relevant to a user's needs. Marking up my primary data with predicative triples would be very time consuming, but it might facilitate the use of NooJ to explore the relationship between person and agency at the macro-level of discourse markers – a development beyond syntactic and semantic features at micro-level.

**References**

Anthony, L. (2019). TagAnt (Version 1.20) [Computer software]. Tokyo, Japan: Waseda University. Retrieved from https://www.laurenceanthony.net/software

Dickinson, E., Johnson, T. H., & Ward, T. V. W. (1958). *The letters of Emily Dickinson*.

Hendler, J., & Berners-Lee, T. (2001). Publishing on the semantic web. *Nature, 410*(6832), 1023-1024. doi:10.1038/35074206

Emily Dickinson Archive. Retrieved from http://www.edickinson.org/

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. Cambridge, Mass: MIT Press.

Harris, Z. S. (1954). Distributional Structure. *WORD, 10*(2-3), 146-162. doi:10.1080/00437956.1954.11659520

Jeffries, L. (2018). Irony in a theory of textual meaning. In M. Jobert & S. Sorlin (Eds.), *The Pragmatics of Irony and Banter* (pp. 23-29). Amsterdam: John Benjamins.

Kurdi, M. Z. (2016). *Natural language processing and computational linguistics: 1, Speech, morphology and syntax*. London, England; Hoboken, New Jersey: iSTE.

Langacker, R. W. (2009). *Investigations in Cognitive Grammar* (D. Geeraerts, R. Dirven, & J. R. Taylor Eds.). Berlin: Mouton de Gruyter.

Louw, B. (1993). Irony in the Text or Insincerity in the Writer? The Diagnostic Potential of Semantic Prosodies. In M. Baker, G. Frances, & E. Tognini-Bonelli (Eds.), *Text And Technology: In Honour Of John Sinclair* (pp. 157-176). Amsterdam: John Benjamins.

McIntryre, D., & Walker, B. (2019). Corpus Stylistics: Theory and Practice. Edinburgh: Edinburgh University Press.

Miller, C. (Ed.) (2016). *Emily Dickinson's Poems As She Preserved Them*. Cambridge, Massachusetts; London: The Belknap Press of Harvard University Press.

Murakami, A., Thompson, P., Hunston, S., & Vajn, D. (2017). 'What is this corpus about?': using topic modelling to explore a specialised corpus. *Corpora, 12*(2), 243-277. doi:10.3366/cor.2017.0118

Silberztein, M. (2019). NooJ: A Linguistic Development Environment (Version 5.1) [Computer software].

Silberztein, M. (2003). *NooJ Manual*. Retrieved from www.nooj4nlp.net

Silberztein, M. (2016). *Formalizing Natural Languages: The NooJ Approach*. London: Wiley.

Simpson, P. (2011). "That's not ironic, that's just stupid": Towards an eclectic account of the discourse of irony. In M. Dynel (Ed.), *The Pragmatics of Humour Across Discourse Domains* (pp. 33-50). Amsterdam: John Benjamins.

Shmid, H. (2019). TreeTagger [Computer software]. University of Munich (LMU). Retrieved from https://cis.uni-muenchen.de/~schmid/tools/TreeTagger/