Towards computation of novel ideas from corpora of scientific text

Haixia Liu[1], James Goulding[2], and Tim Brailsford[1]

[1] School Of Computer Science, University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor Darul Ehsan.
(khyx3lhi,tim.brailsford)@nottingham.edu.my,
[2] Horizon Digital Economy Research, School of Computer Science, University of Nottingham, NG7 2TU, UK.
james.goulding@nottingham.ac.uk

Abstract. In this work we present a method for the computation of novel 'ideas' from corpora of scientific text. The system functions by first detecting concept noun-phrases within the titles and abstracts of publications using Part-Of-Speech tagging, before classifying these into sets of *problem* and *solution* phrases via a target-word matching approach. By defining an idea as a co-occurring <*problem*,*solution*> pair, *known-idea* triples can be constructed through the additional assignment of a relevance value (computed via either phrase co-occurrence or an 'idea frequency-inverse document frequency' score). The resulting triples are then fed into a collaborative filtering algorithm, where problem-phrases are considered as *users* and solution-phrases as the *items* to be recommended. The final output is a ranked list of novel idea candidates, which hold potential for researchers to integrate into their hypothesis generation processes. This approach is evaluated using a subset of publications from the journal *Science*, with precision, recall and F-Measure results for a variety of model parametrizations indicating that the system is capable of generating useful novel ideas in an automated fashion.

Keywords: Idea Mining, Text Mining, Natural Language Processing, Recommender Systems, Collaborative Filtering

1 Introduction

The process of attacking problems by first canvassing participants for spontaneous ideas, collating their responses and distilling the results, is often referred to as *brainstorming*. The term, as popularized by Osborn [26] and expanded upon by Kling [22] and Jessop [19], now corresponds to a well-known set of guidelines for generating creative solutions that entail: discussion of the problem; unconstrained consideration as to how best to solve the problem; screening of the contributions; and, finally, commitment to action. While this approach to problem solving has traditionally required active human participation, in this paper we explore the following challenge: *given the inordinate amount of scientific literature now accessible via the web, is it possible to automate the brainstorming process via machine learning?*

Published in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-23525-7_33

While the idea of supporting the ideation process via technology is not new (the term *Computer-Assisted Brainstorming* was coined three decades ago [17]), prior research has focussed on visualization tools, organizational applications and associated Human-Computer Interaction challenges [14,5,6]. However, text mining and computational linguistic techniques have now progressed to the point that notions of automatically extracting information from text and recognizing the links between underlying topics and concepts has become commonplace [31,2,12]. This brings with it opportunity to not only provide support tools for ideation process, but to actually generate *ideas* themselves.

Generating novel ideas from the automated processing of mass corpora of scientific text requires us to address several conceptual problems. First, we face the issue that the term *'idea'* itself is not at all well-defined from a comprehension perspective [15,20]. Second, new ideas are built upon domain knowledge that is extremely hard, if not impossible, to formalize [37]. Third, ideas from different domains exhibit widely varying characteristics; and finally, commonly used methods for ideation, such as the Gordon technique and expertness [32] are very difficult to computerize. These issues imply that obtaining perfect solutions to problems without human input is unrealistic. However, there is much potential in addressing the sub-task of generating *idea candidates*. Using a functional definition of an "idea" as a $<problem,solution>$ pair (in the vein of [37]), we present an algorithmic approach to idea formulation. Our method breaks the task at hand into the following components: 1. a stage of text mining and linguistic processing of mass scientific corpora; 2. a supervised classification stage to isolate problem and solution concepts; and 3. a stage of re-combination via collaborative filtering, which outputs novel idea pairs for researchers to consider. This approach is evaluated using a subset of publications from the journal *Science*, and both statistical and qualitative evaluations indicate encouraging results. With a corpus of papers that cut across multiple disciplines, it is hoped that some of the idea candidates produced by the system will assist with the sort of cross-disciplinary ideation that is difficult to generate by conventional means.

2 Related Work

The concept of Computer-Assisted Brainstorming (CAB) was established by Hollander [17] in the 1980s, and envisioned interactive computer programs designed to enhance creative thinking. It was several decades later, however, before researchers successfully developed software tools to support brainstorming. Hardenberg *et al.* [14] introduced a *Bare-hand HCI* system, which integrated optical finger tracking into a two-phase brainstorming scenario. Phase 1 involved the collection a large number of ideas from participants and display on a video wall, with phase 2 seeing participants freely and simultaneously rearrange these items via touch manipulation. More recently, Biemann *et al* [5,6] developed *SemanticTalk*, software for visualizing brainstorming sessions and thematic concept trails that acted as a visual memory with both spoken dialogs and text documents being captured on a two-dimensional plane.

One of the key features of *SemanticTalk* was its ability to automatically generate associations between terms within the text identified as being important. This process of identifying key concept terms is being extended by the nascent field of *Idea Mining* [33], which focuses on the task of extracting reified idea structures that are embedded within text - whether that be in websites [42,43], patents [34], databases [27], blogs [35,38] or scientific literature [36]. A general approach for Idea Mining was introduced by Thorleuchter *et al.* in [39], which defined a technological idea as being represented by a combination of a *purpose* and a corresponding *means*, before going on to semi-automatically discover novel idea patterns in unstructured technological texts.

While the systems described above offer valuable digital support for the iteration component in real-world brainstorming, they are all limited in one important respect: they still rely heavily on human input to generate the novel ideas themselves. A possible way to attack this issue is to generate new links between problems and solutions - and a plausible approach to doing this is to harness the success of collaborative filtering (CF) techniques. CF algorithms [7,28] have proven to be extremely effective in generating novel recommendations, both in scientific research and real-world applications. CF uses the known preferences of a group of users to make recommendations (i.e. predictions) of the unknown preferences for other users [30] and CF techniques generally fall into one of three main categories: memory-based, model-based, and hybrid. In this work we focus on the memory-based CF, a method whose most critical component is the mechanism of finding similarities between items and/or users [30]. Many different methods exist to compute similarity [1], and in this work we focus on three that have proven effective in our experiments - log-likelihood, City Block and Tanimoto, all of which are detailed in [13].

Motivated by previous findings in CAB, the idea mining methodology currently being developed in the literature and the established effectiveness of recommender system techniques, we present a new algorithm to generate novel idea candidates. This approach automatically extracts <*problem,solution*> pairs from the titles and abstracts of scientific publications and uses these to computes novel ideas via a CF algorithm. While aimed at helping researchers to conduct scientific research via novel hypothesis generation, our main contribution is to demonstrate the possibility of automating ideation processes via CF techniques.

3 Defining an "Idea"

Young *et al.* [44] describe two principles for producing novel ideas:

- An idea is nothing more or less than a new combination of old elements.
- The capacity to bring old elements into new combinations depends largely on the ability to see relationships.

Based on these principles, we argue that novel idea candidates can be established by uncovering the relationships between *problems* and *solutions* within scientific texts. These components can then be intelligently recombined into previously unforeseen <*problem,solution*> pairs ready for consideration by researchers.

This constructive definition of an "idea" echoes Thorleuchter *et al*'s use of the term, who themselves reference the definitions in [41] in their attempts to identify concepts within various text corpora. In [39] they define an idea as a combination of a *means* and an appertaining *ends*, using unconstrained term vectors to represent each of these entities. In contrast, we represent *problems* and *solutions* using noun-phrases. This assumption is based on previous studies [16,21] which indicate that while a sentence's main conceptual information is usually expressed by both noun- and verb-phrases, its primary concepts are predominantly carried by the latter.

Considering a document T, represented as an ordered set of N words, where $T = \langle w_1, w_2, \ldots, w_N \rangle$, then the functional definitions used to construct our representation of an idea are as follows:

Noun-phrase: A *noun-phrase*, ϕ, is an ordered subset of the text, extracted from T (in our case extracted from the titles or abstracts of publications using part-of-speech tagging technique):

$$\phi = \langle w_1, w_2, \ldots, w_n \rangle. \tag{1}$$

P-phrase: A *p-phrase* is defined as a noun-phrase determined to be a *scientific problem*. We define $\mathcal{P}_T$ as the set of m p-phrases extracted from a document, T (where $m \leq N$):

$$\mathcal{P}_T = \{\phi_a, \phi_b, \phi_c, \ldots\} \tag{2}$$

S-phrase: An *s-phrase* is defined as a noun-phrase that has been categorized as a *technical solution* or a *methodological approach*. We then define $\mathcal{S}_T$ to be the set of q p-phrases extracted from the document, T (where $q \leq N$):

$$\mathcal{S}_T = \{\phi_d, \phi_e, \phi_f, \ldots\}, \textit{where } \mathcal{S}_T \cap \mathcal{P}_T = \emptyset \tag{3}$$

Idea: A specific *idea*[3] can then be defined as a combination of some p-phrase and s-phrase extracted from dataset, D:

$$idea = \langle p\text{–}phrase, s\text{–}phrase \rangle \tag{4}$$

Known Idea: A *known-idea* is defined as a combination of any p-phrase and s-phrase that are found in the same document, T:

$$\exists T \; known\text{–}idea \in \mathcal{P}_T \times \mathcal{S}_T \tag{5}$$

Known-ideas may additionally be attributed a relevance value, representing some measure of the idea's significance within the literature. In this work we evaluate four statistics to measure this significance, described in more detail in §4.5.

Novel Idea: A *novel-idea* is the combination of some p-phrase and s-phrase from the dataset, but which do not co-occur in the same document:

$$\exists T \exists U \; (novel\text{–}idea \in \mathcal{P}_T \times \mathcal{S}_U) \wedge (T \neq U) \tag{6}$$

Novel-ideas may also be assigned a value that reflects the strength of the relationship between its *p-phrase* and *s-phrase* components (as discussed in §4.6).

[3] We of course do not claim that a <problem, solution> pairs represents a universal definition of an idea, but a related pragmatic construct amenable to computation.

4 Methodology

Our method focuses on discovering problem-solution relationships between noun-phrases as detected in the abstracts (together with their titles) of scientific papers. An abstract is a fully self-contained, capsule description of a paper [23]. The noun-phrases it contains should reflect the issue(s) that the author(s) wish to address, so a list of noun-phrases extracted from it provides an ideal foundation for our seed pool of *p-phrases*. If an s-phrase is detected in the same piece of text, semantic relationships between it and neighbouring p-phrases are established[4]. Based on this premise, our approach to subsequently computing novel idea candidates can be broken down into six stages:

1. Noun-phrase extraction from a training-set corpora using Part-Of-Speech tagging.
2. Phrase filtering to remove stop words and text with low information content.
3. Classification of noun-phrases into *p-phrases* (problems) and *s-phrases* (solutions).
4. Aggregation of highly co-occurring <*p-phrase*,*s-phrase*, *relevance*> known-idea triples.
5. Processing of this set of known-idea triples via a collaborative filtering mechanism.
6. Assessment of the resulting ranked list of novel idea candidates that is output.

Several of the steps in this automated process analogise to specific stages in traditional brainstorming sessions. This is demonstrated in Fig. 1, which shows the process of the novel idea computation system on the right, and the corresponding steps in real-world ideation sessions on the left. We examine each of the stages in our method in more detail below.

4.1 Noun-phrase extraction

The first step in our method involves the detection of noun-phrases within the titles and abstracts of the publications that make up our training set. This is undertaken using a standard Part-of-speech (POS) tagging algorithm[5]. While there exist more complex linguistic indicators of an "idea", there are numerous advantages in assuming that noun-phrases are sufficient to represent the informational content of concepts: they are computationally parsimonious; their detection is well understood algorithmically; and studies show that such n-grams preserve far more semantic content than individual term extraction [12]. Recalling our definition of a noun-phrase, $\phi = \langle w_1, w_2, \ldots, w_n \rangle$, for each document, T, we are able to produce a list of noun-phrases: $\Phi_T = \langle \phi_1, \phi_2, ... \rangle$.

[4]For example, consider the sentence *"a dynamic panel data estimation technique is used to examine effects of internal demand on domestic credit"*. The n-gram "dynamic panel data estimation technique" will be recognised as an *s-phrase*, and associated with co-occurring *p-phrases* such as "effects of internal demand on domestic credit").

[5]In this study we have used the *CiteSpace* application for POS tagging [10], which we found performed better than other options such as the *TextBlob* Python library.

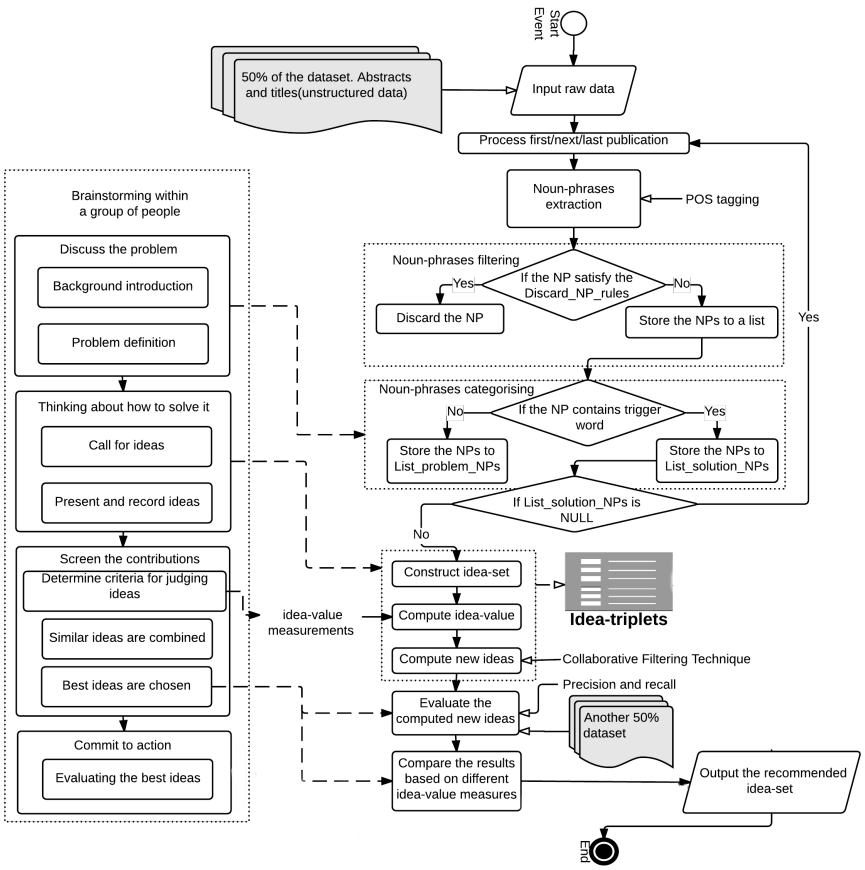

Fig. 1. System flow - each of the method's processing steps are listed on the right hand side, with corresponding stages in real-life brainstorming sessions mapped on the left.

4.2 Noun-phrase filtering

Some noun-phrases generated by the POS tagger are not suitable for inclusion within idea construction. In particular, some concepts will be semantically redundant (i.e. they will have minimal information content in the same vein as *stop words* in traditional informational retrieval tasks). Examples in scientific abstracts are n-grams such as "we present a ***novel model***..." or "our ***general approach*** is tested through an evaluation procedure that...". This stage aims to eliminate such phrases, thus streamlining the method's subsequent processing steps. To this end we employ two filtering steps. First, given a set of hand-crafted "danger" terms W, we remove bi-grams that feature any of its elements:

$$\Phi_{filtered} = \{\phi \in \Phi_T : |\phi| = 2 \rightarrow \phi \cap \mathcal{W} = \varnothing\} \tag{7}$$

The version of $\mathcal{W}$ used in our experiments is listed in Table 7 in the appendix. In addition, we enforce a threshold on the frequency of retained noun-phrases. For this we used Jenks natural breaks classification method [18], assigning p-phrases into five categories according to their frequency across the corpus, eliminating all noun-phrases in the most frequent category. This step is based on the assumption that phrases that exhibit extremely high frequency either have low information content, or reflect noun-phrases that offer no untapped research value.

4.3 Noun-phrases categorization

Now we have a filtered set of noun-phrases we must categorize them into two groups: *p-phrases* and *s-phrases* (representing problems and solutions respectively). There are numerous possible approaches to achieve this task, ranging from *named entity recognition techniques* [25] to the application of *linguistic structure matching* [9]. Unfortunately, in order to utilise these techniques a vast amount of annotated data is required, data that is as-yet-unavailable. Therefore, and in lieu of a fully supervised machine learning approach, we fall back on a rule based pattern-matching approach to identify *s-phrases*. Filtered noun-phrases, $\Phi_{filtered}$, are then compared with a compact bag of trigger words, $\mathcal{G}$, in order to explicitly identify *s-phrases*.

Examples of these cue terms contained in $\mathcal{G}$ might be "method", "approach" and "theory" (the set of trigger words used in our experiments and method of derivation is detailed in Table 6 of the Appendix). Those noun-phrases which remain unmatched are subsequently designated as *p-phrases*[6]. The result is that for a document, T, we produce a set of *s-phrases*, $\mathcal{S}$, and *p-phrases*, $\mathcal{P}$, where:

$$\mathcal{S}_T = \{\phi \in \Phi_{filtered} : \exists w \in \mathcal{G}\,[w \in \phi]\} \tag{8}$$

$$\mathcal{P}_T = \Phi_{filtered} - \mathcal{S}_{\mathcal{T}} \tag{9}$$

This stage of classification has analogies to the real-world brainstorming processes discussed in §2 - in the "discuss the problem" stage of the process [26], if no specific problem angle is specified, participants are instructed to conjure up any noun-phrases that are parts of the problem (i.e. people, places, entities, etc.) in a free-form fashion.

4.4 Known-idea construction

The algorithm must now enumerate *known-ideas* before it will be able to generate novel idea candidates through their recombination. It does this by pairing *p-phrases* and *s-phrases* deemed to be associated with each other. In linguistic processing the specific relation types that are extant between noun-phrases can be uncovered using a range of extraction techniques such as kernel methods; dependency trees [11]; text pattern or structure creation [33]; semantic graphs,

[6] In some ways this is an algorithmic rendition of the arguable expression: "if you are not part of the solution, you are part of the problem".

topic templates and ontologies (e.g. WordNet) [4,40]. However, due to the general qualities of a good abstract [3] - i.e. it should be a condensed and concentrated version of the full text of the research manuscript - we are able to assume that the concepts introduced in a single abstract are all related with each other regarding a specific topic domain. This assumption means we can postulate valid idea-pairs simply by observing the co-occurrence of a problem and solution with the same abstract. We note that this approach may generate some unexpected pairings - this, however, is still in line with the general rules of brainstorming, where the pairing non-obvious components can expand the creativity of a real-world ideation session. The corresponding expansion of the idea pool can increase the chances of producing a radical and effective solution, and as such, we currently neglect some traditional linguistic processing constraints:

1. we do not integrate details of relationship types between noun-phrases.
2. nor distances between the *root* and other nodes in the *Parse Tree*.

Once co-occurring <*p-phrase*,*s-phrase*> pairs have been identified they are assigned a score reflecting their "interestingness" or relevance to the corpus. This value, v, is necessarily subjective, and as such we examine several approaches to determining it, as described in more detail in §4.5. Whatever value is selected, the result of the idea construction process is the set of known-ideas, $\mathcal{K} = \{idea_1, idea_2, ...\}$, as summarised by algorithm 1 below.

Algorithm 1

```
1:  procedure EXTRACT_KNOWN_IDEAS(𝒟)          ▷ 𝒟 is the full document set
2:      𝒦 ←∅                                   ▷ Container for the results
3:      for each T in 𝒟 do
4:          Φ ← extract_noun_phrases(T)
5:          Φ ← filter(Φ)
6:          𝒮_T ← categorize_s-phrases(Φ)
7:          𝒫_T ← categorize_p-phrases(Φ)
8:          for each s in 𝒮_T do
9:              for each p in 𝒫_T do
10:                 v ← compute_idea_value(p, s)
11:                 idea ← ⟨p, s, v⟩
12:                 𝒦 = 𝒦 ∪ {idea}
13:     return 𝒦                               ▷ The output known-idea set
```

4.5 Relevance values for *known-ideas*

In this study we have implemented four statistics which attempt to measure the relevance of a *known-ideas* to future recommendations (and which analogise to the *rating* a user has assigned to an item in traditional collaborative filtering). Each statistic is described below, with examples illustrated in Table 1:

OCC: the simplest way of estimating the relationship intensity between a *p-phrase* and *s-phrase* is to count the number of distinct documents in which they both appear (based on the assumption that the more times they co-occur, the stronger the relationship there is between them):

$$OCC(p,s) = \Big|\{T \in \mathcal{D} : p \in P_T, s \in \mathcal{S}_\mathcal{T}\}\Big| \tag{10}$$

FREQ: idea frequency is similar to document occurrences, but also takes into account the frequency of idea-pair occurrences *within* documents:

$$FREQ(p,s) = \sum_{T \in \mathcal{D}} \Big|\{\langle p,s\rangle \in P_T \times S_T\}\Big| \tag{11}$$

CON: In order to address the fact that term counts alone cannot reflect the fact that some problems have numerous lines of attacks, while others have only a limited solution pool, we have defined the statistic *contribution*. This is a normalization that divides the number of times a certain idea pair co-occurs by the total number of s-phrases used to address the same problem:

$$CON(p,s) = \frac{OCC(p,s)}{\Big|\{T \in \mathcal{D} : s \in S_T\}\Big|} \tag{12}$$

IF-IDF: *idea frequency / inverse document frequency* is an adaptation of the traditional *tf-idf* statistic that we have designed to addresses two observations: 1. the more times an *s-phrase* occurs in any given document, the more likely it is to be a 'key' solution to the document's *p-phrases*, so we wish to favour it; and 2. if an idea-pair crops up across the whole corpus the less likely it is to be "interesting" - either its research value has been saturated, or it is semantically redundant pairing in the same vein as a stop word. IF-IDF balances these two conflicting issues via the following formula[7]:

$$\textit{IF-IDF(p,s)} = \textit{FREQ(p,s)} \times \log\left(\frac{|\mathcal{D}|}{\textit{OCC(p,s)}}\right) \tag{13}$$

Table 1. Examples of known-idea triplets from a sample corpus of 20 documents:

p-phrase	s-phrase	OCC	FREQ	CON	IF-IDF
global warming	climate model	3	24	0.21	19.77
neuropsychiatric disorders	mouse model	3	16	0.50	13.18
impurity atoms	three-dimensional atom probe technique	2	6	0.5	3.00
nickel catalysis	photoredox-metal catalysis approach	1	4	1.00	5.20
impulsive optical excitation	first-principle theoretical simulations	1	3	1.00	3.90

[7] N.b. Idea Frequency (IF) differs from traditional Term Frequency (TF) in that it counts the idea's support over the *whole* corpus, and not just for a single document, resulting in a global statistic.

4.6 Computation of *novel-idea* pairs

We now address the prediction of new links between problems and solutions through comparison of *s-phrase* and *p-phrase* patterns that span across different domains. There are numerous ways to measure similarity between such patterns, but the strategy at the heart of our techniques is based upon a collaborative filtering [7]. Collaborative filtering and the recommendation systems they underpin are based on imputation - a target user's past behaviours are first modelled and then compared to the habits of other users. Items favoured by similar users, but which do not yet exist in the target user's history, are then used as the basis for new recommendations. Our technique considers *p-phrases* as analogs to users, and *s-phrases* as items. Traditional recommender systems can be formulated as *user-based* or *item-based* algorithms [28] and we assess both approaches. In the generation of novel-ideas, our collaborative filtering task consists of the following steps:

1. **construct a preference matrix:** in our case, each row represents a *p-phrase* and each column represents an *s-phrase* with the numerical value at the intersection of a row and a column represents the idea's relevance value, v (as selected from one of the statistics in §4.4).
2. **compute similarity scores**: for a specific problem vector (i.e. a row in the preference matrix), u, iterate through every other problem vector, w, and compute a similarity s between u and w and retain the k nearest neighbours, $\mathcal{N}$. In our experiments we optimize N for each of the following distance metrics: *Tanimoto*, *Loglikelihood* and *CityBlock*[8].
3. **generate novel-idea pairs:** this is achieved by recommending novel solutions to existing problems. For each potential solution, i, that the current problem has no entry for, we consider every vector, w, in the neighbourhood, N, and add its relevance score for solution i to a running average, weighted by the vector's similarity score s. Finally, results are sorted, producing is a ranked list of novel *s-phrases* to the *p-phrase* under consideration. The top n *s-phrases* are combined with the p-phrase under consideration as our novel-idea prediction (in our experiments n is drawn from 2, 5, 10).

5 Experimental Evaluation

A collection of the titles and abstracts was studied, extracted randomly from 3072 English language articles published in the journal *Science*. The dataset, covering the years 1998-2015, was partitioned so that half of the articles formed our training set, $\mathcal{D}$, and the other half our test set. After noun-phrase filtering, Φ contained 48,958 noun-phrases and noun-phrase categorization resulted in 46,035 p-phrases and 2,923 s-phrases. From this the algorithm constructed 70,760 unique <*p-phrase*,*s-phrase*> known-idea pairs[9].

[8] please refer to http://mahout.apache.org for implementation details.

[9] The restricted number of known-ideas is because no cogent s-phrases could be extracted for many abstracts, even though numerous noun-phrases were identified.

For each model parametrization a set of novel idea candidates were generated from the training set. These were then evaluated to determine if recommended idea-candidates *actually occurred* in the test set, with results being summarized for each *p-phrase* using traditional precision, recall and F-measure scores. This process was repeated for both user- and item-based collaborative filtering, using relevance value statistics drawn from {OCC, CON, IF-IDF} (n.b. FREQ is not reported due to its similarity to OCC) and varying the size of the recommended solution list for each p-phrase ($n \in \{2, 5, 10\}$). Overall mean precision, recall and F-measure scores were produced for each of the 54 model parameterizations (we report the results for each model using an optimized neighbourhood).

5.1 Results

Every stage of our approach has the potential for future refinement. Despite this, the novel-idea candidates output by the system's first iteration were highly encouraging. Examples of the system's output in Table 2, taken from a range of categories in the Science corpus, illustrate the cogent recommendations the system can produce. Precision and recall results are similarly positive - full results in Tables 3-5 indicate how the number of items in the recommendation set influences results for each of the relevance values we tested (with the size of the neighbourhood being optimized for each recommendation set size).

Table 2. Examples of novel idea-pairs generated from, using *User-CityBlock* similarity and OCC metric, with a neighbourhood of $k = 50$.

problem	old-solution	proposed-solution
first stars	cosmological simulations	nucleosynthesis model
creep damage	diffraction analysis	thermodynamic analysis
ancestral state reconstruction	likelihood-based approach	fluorescence technique
yeast genes	biochemical genomics approach	phenotypic analysis
heritable functional states	recombinant method	neuronal differentiation

Because we are assessing the efficacy of our idea recommendation approach as a whole rather than contrasting results for different collaborative filtering parameterizations, let us first consider the system's top *2-recommendations*. The results tables illustrate that across the board the system's top two novel idea recommendations match our test set over 90% of the time (with a maximum recall of 0.941 when using the CityBlock similarity measure and a relevance value based on OCC - see Table 2 for example idea pairs). While these statistical results are highly encouraging we note that extensive human evaluation of output ideas is required before we can be confident that these results could be translated into hypothesis generation processes. Additionally - and as one might expect - as the size of our recommendation list increases, results drop off starkly (by the time we have reached 10-recommendations the F-measure of our recommendations has fallen by almost half). This indicates that the system currently works optimally only for its highest ranked recommendations.

Table 3. OCC performance (precision/recall/F-measure)

metric	2-recommendations	5-recommendations	10-recommendations
user Loglikelihood	0.900/0.928/0.913	0.557/0.734/0.633	0.374/0.527/0.437
user CityBlock	0.951/0.941/0.946	0.730/0.765/0.747	0.474/0.685/0.560
user Tanimoto	0.901/0.929/0.915	0.561/0.735/0.636	0.366/0.536/0.435
item Loglikelihood	0.606/0.635/0.620	0.480/0.716/0.575	0.374/0.824/0.515
item CityBlock	0.209/0.208/0.208	0.027/0.027/0.027	0.013/0.126/0.023
item Tanimoto	0.423/0.437/0.430	0.259/0.403/0.315	0.267/0.611/0.372

Table 4. CON performance (precision/recall/F-measure)

metric	2-recommendations	5-recommendations	10-recommendations
user Loglikelihood	0.890/0.926/0.908	0.557/0.743/0.637	0.401/0.591/0.478
user CityBlock	0.943/0.939/0.941	0.730/0.761/0.745	0.493/0.674/0.570
user Tanimoto	0.892/0.928/0.910	0.562/0.747/0.641	0.393/0.600/0.475
item Loglikelihood	0.601/0.637/0.618	0.486/0.713/0.578	0.414/0.840/0.554
item CityBlock	0.208/0.210/0.209	0.034/0.050/0.040	0.027/0.135/0.045
item Tanimoto	0.422/0.443/0.432	0.267/0.407/0.323	0.305/0.655/0.416

Table 5. IF-IDF performance (precision/recall/F-measure)

metric	2-recommendations	5-recommendations	10-recommendations
user Loglikelihood	0.890/0.926/0.908	0.600/0.743/0.639	0.401/0.591/0.478
user CityBlock	0.943/0.940/0.941	0.730/0.761/0.745	0.493/0.674/0.570
user Tanimoto	0.892/0.928/0.910	0.565/0.747/0.643	0.393/0.600/0.475
item Loglikelihood	0.602/0.637/0.619	0.489/0.713/0.580	0.414/0.840/0.554
item CityBlock	0.207/0.210/0.210	0.034/0.050/0.041	0.027/0.135/0.045
item Tanimoto	0.422/0.443/0.432	0.267/0.407/0.323	0.305/0.655/0.416

In a comparison of the distance metric used to determine CF neighbourhoods, the *CityBlock* measure is the clear winner. This represents absolute distance between solution vectors, and for all parameterizations of the model it consistently returns the highest F-measure results (this is down mostly to its superior precision results, with recall being relatively consistent across all distance measures).

A clear contrast also exists between results for user- and item-based collaborative filtering approaches, with the former performing far better than the latter in all cases. We conclude from these results that it is far better to recommend new solutions to old problems, than to try and bring new problems to old solutions. In many ways this is an intuitive result, as it is far more likely that extant solutions will be immediately attempted when new research problems arise.

Finally we consider the effectiveness of the three idea relevance scores tested. Despite being the least complex statistic implemented, OCC provides the strongest results. While only edging out the other statistics tested, the preference for OCC is statistically significant (with a paired t-test producing a p-value of < 0.001). Results for CON and IF-IDF are almost indistinguishable, and examination of idea recommendations for each problem indicate an extremely high crossover (in fact 44% of problems received identical recommendation sets for all sizes of recommendation list). These results appear to indicate that simply counting idea-pair occurrences in the dataset is a sufficient basis to assess the significance of a solution to any given problem.

6 Discussion

This study demonstrated the plausibility of generating novel idea-candidates in an automated fashion. User-based CF offered the best performance and, while different distance measures produced comparable results, OCC provided a simple method to achieve the most effective performance. Nonetheless, there is scope for further research at each of the stages of the idea-generation process.

First, there is potential to improve the filtering of noun-phrases identified by POS tagging (based perhaps upon a more formalized information-theoretic approach to detecting 'semantically redundant' terms). Second, our approach to classifying s-phrases and p-phrases remains relatively coarse, using a pattern matching approach based on trigger words. A further investigation of this processing stage would be of particular interest and numerous options seem viable.

It is our aim, for example, to implement a supervised classification model to improve detection of *s-phrases* and *p-phrases*. The input features for this model could be generated from language models [29], lexical cohesion [24] and linguistic grammar-based techniques [8], in addition to the statistical features already used. Training would need to be performed on ground truth annotations of scientific abstracts, but these could be collated in a crowd-sourced fashion by presenting abstracts to domain experts and allowing them to manually identify problem and solution term patterns within the text. The goal here would be to directly address some of the limitations with our current approach, such as the fact that *p-phrases* and *s-phrases* are overly dependent upon their context (for example, a p-phrase in one document might be an s-phrase in another).

Additional areas of interest lie not only in investigating other similarity measures from the collaborative filtering literature, but also in exploring other external indicators of a known-idea's relevance value. These might include the number of citations generated by the paper the idea appears in, or the impact factor of its parent publication, or indeed any of the host of methods that are used to assess the relevance of a paper within the scientific literature.

Finally, and perhaps of greatest importance, there is a good deal of room to extend our evaluation of the efficacy of the ideas generated by the system. Currently, we assess a novel-idea candidate's merit based upon whether it occurs in (or is absent from) future literature. This neglects two factors: 1. the comprehensibility and interestingness of generated idea-pairs (a situation which can only be addressed by a programme of human evaluation of the system's outputs), and 2. any assessment of an idea's *inventiveness*. Currently, if a recommended idea does not appear in our test set, it is deemed as a false positive out of hand, whereas it may be the case that the idea is simply yet to be researched.

7 Conclusion

In this study, we have presented a first approach for generating novel idea candidates from corpora of scientific text, that is decomposable into six distinct stages. Noun-phrases are extracted from the abstracts of scientific papers via POS tagging; a filtering process occurs to remove redundant concepts; the results set

of phrases are subsequently categorized into problem and solution; co-occurring pairs are assigned a relevance score (based on number of co-occurrences, contribution to a problem's overall support or an idea frequency/inverse document frequency score); and finally a collaborative filtering algorithm generates new idea recommendations. This process illustrates the ability to transform of unstructured textual data into structured idea pairs, and the potential to manipulate that structure computationally to generate new idea candidates. The approach was evaluated using a subset of publications from the journal *Science*, and both statistical and qualitative evaluations indicate strongly encouraging results, with an OCC relevance value combined with a (user-based) CityBlock similarity measure offering the best performance. Our hope is that in establishing this modular approach to automated idea generation, each stage may be honed by the broader research community to ultimately produce a system that has real utility to hypothesis generation.

8 Acknowledgments

This work was jointly supported by CFFRC-PLUS PhD scholarship scheme, the RCUK Horizon Digital Economy Research Hub grant, EP/G065802/1 and the EPSRC Neodemographics grant, EP/L021080/1, .

References

1. Ahn, H.J.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences 178(1), 37–51 (2008)
2. Allan, J., Carbonell, J.G., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study final report (1998)
3. Andrade, C.: How to write a good abstract for a scientific paper or conference presentation. Indian journal of psychiatry 53(2), 172 (2011)
4. Banko, M., Etzioni, O., Center, T.: The tradeoffs between open and traditional relation extraction. In: ACL. vol. 8, pp. 28–36. Citeseer (2008)
5. Biemann, C., Böhm, K., Heyer, G., Melz, R.: Semantictalk: Software for visualizing brainstorming sessions and thematic concept trails on document collections. In: Knowledge Discovery in Databases: PKDD 2004, pp. 534–536. Springer (2004)
6. Biemann, C., Böhm, K., Heyer, G., Melz, R.: Automatically building concept structures and displaying concept trails for the use in brainstorming sessions. In: Innovative Internet Community Systems, pp. 157–167. Springer (2006)
7. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. pp. 43–52. Morgan Kaufmann (1998)
8. Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Comput. Linguist. 18(4), 467–479 (Dec 1992), http://dl.acm.org/citation.cfm?id=176313.176316
9. Bybee, J.L., Hopper, P.J.: Frequency and the emergence of linguistic structure, vol. 45. John Benjamins Publishing (2001)
10. Chen, C.: Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. Journal of the American Society for information Science and Technology 57(3), 359–377 (2006)

11. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. p. 423. Association for Computational Linguistics (2004)
12. Ding, W., Chen, C.: Dynamic topic detection and tracking: A comparison of hdp, c-word, and cocitation methods. Journal of the Association for Information Science and Technology (2014)
13. Guo, S., et al.: Analysis and evaluation of similarity metrics in collaborative filtering recommender system (2014)
14. von Hardenberg, C., Bérard, F.: Bare-hand human-computer interaction. In: Proceedings of the 2001 Workshop on Perceptive User Interfaces. pp. 1–8. PUI '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/971478.971513
15. Hare, V.C., Milligan, B.: Main idea identification: Instructional explanations in four basal reader series. Journal of Literacy Research 16(3), 189–204 (1984)
16. Hildreth, P.M., Kimble, C.: Knowledge networks: Innovation through communities of practice. IGI Global (2004)
17. Hollander, S.: Computer-assisted Creativity and the Policy Process. Thayer School of Engineering (1984), http://books.google.com.my/books?id=qEiLtgAACAAJ
18. Jenks, G.F.: The data model concept in statistical mapping. International yearbook of cartography 7(1), 186–190 (1967)
19. Jessop, J.L.: Expanding our students' brainpower: Idea generation and critical thinking skills. Antennas and Propagation Magazine, IEEE 44(6), 140–144 (2002)
20. Jitendra, A.K., Cole, C.L., Hoppes, M.K., Wilson, B.: Effects of a direct instruction main idea summarization program and self-monitoring on reading comprehension of middle school students with learning disabilities. Reading & Writing Quarterly: Overcoming Learning Difficulties 14(4), 379–396 (1998)
21. Kamp, H.: A theory of truth and semantic representation. Formal semantics-the essential readings pp. 189–222 (1981)
22. Kling, H.: Get more out of group projects by using structured brainstorming. QUALITY PROGRESS 23(3), 136–136 (1990)
23. Koopman, P.: How to write an abstract. Carnegie Mellon University. Retrieved May 31, 2013 (1997)
24. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational linguistics 17(1), 21–48 (1991)
25. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
26. Osborn, A.F.: Applied imagination. (1953)
27. Park, Y., Lee, S.: How to design and utilize online customer center to support new product concept generation. Expert Systems with Applications 38(8) (2011)
28. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms pp. 285–295 (2001)
29. Song, F., Croft, W.B.: A general language model for information retrieval. In: Proceedings of the eighth international conference on Information and knowledge management. pp. 316–321. ACM (1999)
30. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in artificial intelligence 2009, 4 (2009)
31. Tan, A.H., et al.: Text mining: The state of the art and the challenges. In: Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases. pp. 65–70 (1999)
32. Taylor, J.W.: How to create new ideas. Prentice-Hall (1961)

33. Thorleuchter, D.: Finding new technological ideas and inventions with text mining and technique philosophy. In: Data Analysis, Machine Learning and Applications, pp. 413–420. (2008)
34. Thorleuchter, D., den Poel, D.V., Prinzie, A.: A compared r&d-based and patent-based cross impact analysis for identifying relationships between technologies. Technological Forecasting and Social Change 77(7), 1037–1050 (2010)
35. Thorleuchter, D., Van den Poel, D.: Companies website optimising concerning consumer's searching for new products. In: Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on. vol. 1. IEEE (2011)
36. Thorleuchter, D., Van den Poel, D.: Semantic technology classificationa defence and security case study. In: Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on. vol. 1, pp. 36–39. IEEE (2011)
37. Thorleuchter, D., Van den Poel, D.: Extraction of ideas from microsystems technology. In: Advances in Computer Science and Information Engineering, (2012)
38. Thorleuchter, D., Van den Poel, D., Prinzie, A.: Extracting consumers needs for new products-a web mining approach. In: Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on. pp. 440–443. IEEE (2010)
39. Thorleuchter, D., den Poel, D.V., Prinzie, A.: Mining ideas from textual information. Expert Systems with Applications 37(10), 7182–7188 (2010)
40. Trampuš, M., Mladenic, D.: Constructing domain templates with concept hierarchy as background knowledge. Information Technology And Control 43(4) (2014)
41. Wallas, G.: The art of thought. (1926)
42. Wang, C., Lu, J., Zhang, G.: Mining key information of web pages: A method and its application. Expert Systems with Applications 33(2), 425–433 (2007)
43. Yoon, J.: Detecting weak signals for long-term business opportunities using text mining of web news. Expert Systems with Applications 39(16), 12543–12550 (2012)
44. Young, J.W.: A technique for producing ideas. NTC Business Books (1975)

A APPENDIX

Table 6. S-phrase cue terms - "method" was used as a seed term, with trigger words being expanded through synonym extraction via *www.thesaurus.com* and isolating nearest neighbours using *Word2vec* (see https://code.google.com/p/word2vec/).

approach, technique, scheme, algorithm, analysis,modelling,methodology, strategy, framework, tool, procedure, structure, processing, heuristic, mechanism, architecture, theory, paradigm, formalism, platform

Table 7. Noun-Phrase filtering terms

overall, primary, key, valuable, excellent, potential, essential, unique, numerous, important, prior, practical, basic, different, simple, successful, current, possible, previous, existing, well-established, independent, particular, usual, new, old, powerful, main, common, detailed, efficient, good, acceptable, effective, novel, state-of-the-art, useful, modern, unreliable, additional, methodological, available, recent, general, specific, creative, brief, critical, major, second, reasonable, various, personal, latest , interesting