A Bootstrapping Approach to Unsupervised Detection of Cue Phrase Variants

Rashid M. Abdalla and Simone Teufel
Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue, Cambridge CB3 OFD, UK
rma33@cam.ac.uk, sht25@cam.ac.uk

Abstract

We investigate the unsupervised detection of semi-fixed cue phrases such as "*This paper proposes a novel approach...*[1]" from unseen text, on the basis of only a handful of seed cue phrases with the desired semantics. The problem, in contrast to bootstrapping approaches for Question Answering and Information Extraction, is that it is hard to find a constraining context for occurrences of semi-fixed cue phrases. Our method uses components of the cue phrase itself, rather than external context, to bootstrap. It successfully excludes phrases which are different from the target semantics, but which look superficially similar. The method achieves 88% accuracy, outperforming standard bootstrapping approaches.

1 Introduction

Cue phrases such as "*This paper proposes a novel approach to...*", "*no method for ... exists*" or even "*you will hear from my lawyer*" are semi-fixed in that they constitute a formulaic pattern with a clear semantics, but with syntactic and lexical variations which are hard to predict and thus hard to detect in unseen text (e.g. "*a new algorithm for ... is suggested in the current paper*" or "*I envisage legal action*"). In scientific discourse, such *meta-discourse* (Myers, 1992; Hyland, 1998) abounds and plays an important role in marking the discourse structure of the texts.

Finding these variants can be useful for many text understanding tasks because semi-fixed cue phrases act as linguistic markers indicating the importance and/or the rhetorical role of some adjacent text. For the summarisation of scientific papers, cue phrases such as "*Our paper deals with...*" are commonly used as indicators of extraction-worthiness of sentences (Kupiec et al., 1995). Re-generative (rather than extractive) summarisation methods may want to go further than that and directly use the knowledge that a certain sentence contains the particular research aim of a paper, or a claimed gap in the literature. Similarly, in the task of automatic routing of customer emails and automatic answering of some of these, the detection of threats of legal action could be useful.

However, systems that use cue phrases usually rely on manually compiled lists, the acquisition of which is time-consuming and error-prone and results in cue phrases which are genre-specific. Methods for finding cue phrases automatically include Hovy and Lin (1998) (using the ratio of word frequency counts in summaries and their corresponding texts), Teufel (1998) (using the most frequent n-grams), and Paice (1981) (using a pattern matching grammar and a lexicon of manually collected equivalence classes). The main issue with string-based pattern matching techniques is that they cannot capture syntactic generalisations such as active/passive constructions, different tenses and modification by adverbial, adjectival or prepositional phrases, appositions and other parenthetical material.

For instance, we may be looking for sentences expressing the goal or main contribution of a paper; Fig. 1 shows candidates of such sentences. Cases a)–e), which do indeed describe the authors' goal, display a wide range of syntactic variation.

a) *In this paper, we* ***introduce*** *a* ***method*** *for similarity-based estimation of...*
b) *We* ***introduce*** *and justify a* ***method****...*
c) *A* ***method*** *(described in section 1) is* ***introduced***
d) *The* ***method introduced*** *here is a variation...*
e) *We wanted to* ***introduce*** *a* ***method****...*
f) *We do not* ***introduce*** *a* ***method****...*
g) *We* ***introduce*** *and adopt the* ***method*** *given in [1]...*
h) *Previously we* ***introduced*** *a similar* ***method****...*
i) *They* ***introduce*** *a similar* ***method****...*

Figure 1: Goal statements and syntactic variation – correct matches (a-e) and incorrect matches (f-i)

[1] In contrast to standard work in discourse linguistics, which mostly considers sentence connectives and adverbials as cue phrases, our definition includes longer phrases, sometimes even entire sentences.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 921–928,
Sydney, July 2006.

Cases f)–i) in contrast are false matches: they do not express the authors' goals, although they are superficially similar to the correct contexts. While string-based approaches (Paice, 1981; Teufel, 1998) are too restrictive to cover the wide variation within the correct contexts, bag-of-words approaches such as Agichtein and Gravano's (2000) are too permissive and would miss many of the distinctions between correct and incorrect contexts.

Lisacek et al. (2005) address the task of identifying "paradigm shift" sentences in the biomedical literature, i.e. statements of thwarted expectation. This task is somewhat similar to ours in its definition by rhetorical context. Their method goes beyond string-based matching: In order for a sentence to qualify, the right set of concepts must be present in a sentence, with any syntactic relationship holding between them. Each concept set is encoded as a fixed, manually compiled lists of strings. Their method covers only one particular context (the paradigm shift one), whereas we are looking for a method where many types of cue phrases can be acquired. Whereas it relies on manually assembled lists, we advocate data-driven acquisition of new contexts. This is generally preferrable to manual definition, as language use is changing, inventive and hard to predict and as many of the relevant concepts in a domain may be infrequent (cf. the formulation "*be cursed*", which was used in our corpus as a way of describing a method's problems). It also allows the acquisition of cue phrases in new domains, where the exact prevalent meta-discourse might not be known.

Riloff's (1993) method for learning information extraction (IE) patterns uses a syntactic parse and correspondences between the text and filled MUC-style templates to learn context in terms of lexico-semantic patterns. However, it too requires substantial hand-crafted knowledge: 1500 filled templates as training material, and a lexicon of semantic features for roughly 3000 nouns for constraint checking. Unsupervised methods for similar tasks include Agichtein and Gravano's (2000) work, which shows that clusters of vector-space-based patterns can be successfully employed to detect specific IE relationships (companies and their headquarters), and Ravichandran and Hovy's (2002) algorithm for finding patterns for a Question Answering (QA) task. Based on training material in the shape of pairs of question and answer terms – e.g., (e.g. {*Mozart, 1756*}), they learn the

a) *In this paper, we* ***introduce*** *a* ***method*** *for similarity-based estimation of . . .*
b) *Here, we* ***present*** *a similarity-based* ***approach*** *for estimation of. . .*
c) *In this paper, we* ***propose*** *an* ***algorithm*** *which is . . .*
d) *We will here* ***define*** *a* ***technique*** *for similarity-based. . .*

Figure 2: Context around cue phrases (lexical variants)

semantics holding between these terms ("birth year") via frequent string patterns occurring in the context, such as "*A was born in B*", by considering n-grams of all repeated substrings. What is common to these three works is that bootstrapping relies on constraints between the context *external* to the extracted material and the extracted material itself, and that the target extraction material is defined by real-world relations.

Our task differs in that the cue phrases we extract are based on general rhetorical relations holding in all scientific discourse. Our approach for finding semantically similar variants in an unsupervised fashion relies on bootstrapping of seeds from *within the cue phrase*. The assumption is that every semi-fixed cue phrase contains at least two main concepts whose syntax and semantics mutually constrain each other (e.g. verb and direct object in phrases such as "*(we) present an approach for*"). The expanded cue phrases are recognised in various syntactic contexts using a parser[2]. General semantic constraints valid for groups of semantically similar cue phrases are then applied to model, e.g., the fact that it must be the authors who present the method, not somebody else.

We demonstrate that such an approach is more appropriate for our task than IE/QA bootstrapping mechanisms based on cue phrase-external context. Part of the reason for why normal bootstrapping does not work for our phrases is the difficulty of finding negatives contexts, essential in bootstrapping to evaluate the quality of the patterns automatically. IE and QA approaches, due to uniqueness assumptions of the real-world relations that these methods search for, have an automatic definition of *negative* contexts by hard constraints (i.e., all contexts involving Mozart and any other year are by definition of the wrong semantics; so are all contexts involving Microsoft and a city other than Redmond). As our task is not grounded in real-world relations but in rhetorical ones, constraints found in the context tend to be

[2]Thus, our task shows some parallels to work in paraphrasing (Barzilay and Lee, 2002) and syntactic variant generation (Jacquemin et al., 1997), but the methods are very different.

soft rather than hard (cf. Fig 2): while it it possible that strings such as "*we*" and "*in this paper*" occur more often in the context of a given cue phrase, they also occur in many other places in the paper where the cue phrase is not present. Thus, it is hard to define clear negative contexts for our task.

The novelty of our work is thus the new pattern extraction task (finding variants of semi-fixed cue phrases), a task for which it is hard to directly use the context the patterns appear in, and an iterative unsupervised bootstrapping algorithm for lexical variants, using phrase-internal seeds and ranking similar candidates based on relation strength between the seeds.

While our method is applicable to general cue phrases, we demonstrate it here with transitive verb–direct object pairs, namely a) cue phrases introducing a new methodology (and thus the main research goal of the scientific article; e.g. "*In this paper, we propose a novel algorithm...*") – we call those *goal-type* cue phrases; and b) cue phrases indicating continuation of previous other research (e.g. "*Therefore, we adopt the approach presented in [1]...*") – *continuation-type* cue phrases.

2 Lexical Bootstrapping Algorithm

The task of this module is to find lexical variants of the components of the seed cue phrases. Given the seed phrases "*we introduce a method*" and "*we propose a model*", the algorithm starts by finding all direct objects of "*introduce*" in a given corpus and, using an appropriate similarity measure, ranks them according to their distributional similarity to the nouns "*method*" and "*model*". Subsequently, the noun "*method*" is used to find transitive verbs and rank them according to their similarity to "*introduce*" and "*propose*". In both cases, the ranking step retains variants that preserve the semantics of the cue phrase (e.g. "*develop*" and "*approach*") and filters irrelevant terms that change the phrase semantics (e.g. "*need*" and "*example*").

Stopping at this point would limit us to those terms that co-occur with the seed words in the training corpus. Therefore additional iterations using automatically generated verbs and nouns are applied in order to recover more and more variants. The full algorithm is given in Fig. 3.

The algorithm requires corpus data for the steps Hypothesize (producing a list of potential candidates) and Rank (testing them for similarity). We

Input: Tuples $\{A_1, A_2, \ldots, A_m\}$ and $\{B_1, B_2, \ldots, B_n\}$.
Initialisation: Set the concept-A reference set to $\{A_1, A_2, \ldots, A_m\}$ and the concept-B reference set to $\{B_1, B_2, \ldots, B_n\}$. Set the concept-A active element to A_1 and the concept-B active element to B_1.
Recursion:

1. **Concept B retrieval:**
   (i) **Hypothesize:** Find terms in the corpus which are in the desired relationship with the concept-A active element (e.g. direct objects of a verb active element). This results in the concept-B candidate set.
   (ii) **Rank:** Rank the concept-B candidate set using a suitable ranking methodology that may make use of the concept-B reference set. In this process, each member of the candidate set is assigned a score.
   (iii) **Accumulate:** Add the top s items of the concept-B candidate set to the concept-B accumulator list (based on empirical results, s is the rank of the candidate set during the initial iteration and 50 for the remaining iterations). If an item is already on the accumulator list, add its ranking score to the existing item's score.
2. **Concept A retrieval:** as above, with concepts A and B swapped.
3. **Updating active elements:**
   (i) Set the concept-B active element to the highest ranked instance in the concept-B accumulator list which has not been used as an active element before.
   (ii) Set the concept-A active element to the highest ranked instance in the concept-A accumulator list which has not been used as an active element before.

Repeat steps 1-3 for k iterations
Output: top M words of concept-A (verb) accumulator list and top N words of concept-B (noun) accumulator list

Reference set: a set of seed words which define the collective semantics of the concept we are looking for in this iteration
Active element: the instance of the concept used in the current iteration for retrieving instances of the other concept. If we are finding lexical variants of Concept A by exploiting relationships between Concepts A and B, then the active element is from Concept B.
Candidate set: the set of candidate terms for one concept (eg. Concept A) obtained using an active element from the other concept (eg. Concept B). The more semantically similar a term in the candidate set is to the members of the reference set, the higher its ranking should be. This set contains verbs if the active element is a noun and vice versa.
Accumulator list: a sorted list that accumulates the ranked members of the candidate set.

Figure 3: Lexical variant bootstrapping algorithm

estimate frequencies for the Rank step from the written portion of the British National Corpus (BNC, Burnard (1995)), 90 Million words. For the Hypothesize step, we experiment with two data sets: First, the scientific subsection of the BNC (24 Million words), which we parse using RASP (Briscoe and Carroll, 2002); we then examine the grammatical relations (GRs) for transitive verb constructions, both in active and passive voice. This method guarantees that we find almost all transitive verb constructions cleanly; Carroll et al. (1999) report an accuracy of .85 for

DOs, Active: "AGENT_STRING AUX active-verb-element DETERMINER * POSTMOD"
DOs, Passive: "DETERMINER * AUX active-verb-element element"
TVs, Active: "AGENT_STRING AUX * DETERMINER active-noun- element POSTMOD"
TVs, Passive:"DET active-noun-element AUX * POSTMOD"

Figure 4: Query patterns for retrieving direct objects (DOs) and transitive verbs (TVs) in the Hypothesize step.

newspaper articles for this relation. Second, in order to obtain larger coverage and more current data we also experiment with Google Scholar[3], an automatic web-based indexer of scientific literature (mainly peer-reviewed papers, technical reports, books, pre-prints and abstracts). Google Scholar snippets are often incomplete fragments which cannot be parsed. For practical reasons, we decided against processing the entire documents, and obtain an approximation to direct objects and transitive verbs with regular expressions over the result snippets in both active and passive voice (cf. Fig. 4), designed to be high-precision[4]. The amount of data available from BNC and Google Scholar is not directly comparable: harvesting Google Scholar snippets for both active and passive constructions gives around 2000 sentences per seed (Google Scholar returns up to 1000 results per query), while the number of BNC sentences containing seed words in active and passive form varies from 11 ("*formalism*") to 5467 ("*develop*") with an average of 1361 sentences for the experimental seed pairs.

Ranking

Having obtained our candidate sets (either from the scientific subsection of the BNC or from Google Scholar), the members are ranked using BNC frequencies. We investigate two ranking methodologies: frequency-based and context-based. Frequency-based ranking simply ranks each member of the candidate set by how many times it is retrieved together with the current active element. Context-based ranking uses a similarity measure for computing the scores, giving a higher score to those words that share sufficiently similar contexts with the members of the reference set. We consider similarity measures in a vector space defined either by a fixed window, by the sentence window, or by syntactic relationships. The score assigned to each word in the candidate set is the sum of its semantic similarity values computed with respect to each member in the reference set. Syntactic contexts, as opposed to window-based contexts, constrain the context of a word to only those words that are grammatically related to it. We use verb-object relations in both active and passive voice constructions as did Pereira et al. (1993) and Lee (1999), among others. We use the cosine similarity measure for window-based contexts and the following commonly used similarity measures for the syntactic vector space: Hindle's (1990) measure, the weighted Lin measure (Wu and Zhou, 2003), the α-Skew divergence measure (Lee, 1999), the Jensen-Shannon (JS) divergence measure (Lin, 1991), Jaccard's coefficient (van Rijsbergen, 1979) and the Confusion probability (Essen and Steinbiss, 1992). The Jensen-Shannon measure $\mathrm{JS}(\mathrm{x}_1, \mathrm{x}_2) = \sum_{\mathrm{y}\in\mathrm{Y}} \sum_{\mathrm{x}\in\{\mathrm{x}_1,\mathrm{x}_2\}} \left(\mathrm{P}(\mathrm{y}|\mathrm{x}) \log \left(\frac{\mathrm{P}(\mathrm{y}|\mathrm{x})}{\frac{1}{2}(\mathrm{P}(\mathrm{y}|\mathrm{x}_1)+\mathrm{P}(\mathrm{y}|\mathrm{x}_2))}\right)\right)$ subsequently performed best for our task. We compare the different ranking methodologies and data sets with respect to a manually-defined gold standard list of 20 goal-type verbs and 20 nouns. This list was manually assembled from Teufel (1999); WordNet synonyms and other plausible verbs and nouns found via Web searches on scientific articles were added. We ensured by searches on the ACL anthology that there is good evidence that the gold-standard words indeed occur in the right contexts, i.e. in goal statement sentences. As we want to find similarity metrics and data sources which result in accumulator lists with many of these gold members at high ranks, we need a measure that rewards exactly those lists. We use non-interpolated Mean Average Precision (MAP), a standard measure for evaluating ranked information retrieval runs, which combines precision and recall and ranges from 0 to 1[5].

We use 8 pairs of 2-tuples as input (e.g. [*introduce, study*] & [*approach*, *method*]), randomly selected from the gold standard list. MAP was cal-

[3] http://scholar.google.com

[4] The capitalised words in these patterns are replaced by actual words (e.g. AGENT_STRING: We/I, DETERMINER: a/ an/our), and the extracted words (indicated by "*") are lemmatised.

[5] $MAP = \frac{1}{N}\sum_{j=1}^{N} AP_j = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{M}\sum_{i=1}^{M} P(g_i)$ where $P(g_i) = \frac{n_{ij}}{r_{ij}}$ if g_i is retrieved and 0 otherwise, N is the number of seed combinations, M is the size of the golden list, g_i is the i^th member of the golden list and r_{ij} is its rank in the retrieved list of combination j while n_{ij} is the number of golden members found up to and including rank r_{ij}.

Ranking scheme	BNC	Google Scholar
Frequency-based	0.123	0.446
Sentence-window	0.200	0.344
Fixedsize-window	0.184	0.342
Hindle	0.293	0.416
Weighted Lin	0.358	0.509
α-Skew	0.361	0.486
Jensen-Shannon	0.404	0.550
Jaccard's coef.	0.301	0.436
Confusion prob.	0.171	0.293

Figure 5: MAPs after the first iteration

culated over the verbs and nouns retrieved using our algorithm and averaged. Fig. 5 summarises the MAP scores for the first iteration, where Google Scholar significantly outperformed the BNC. The best result for this iteration (MAP=.550) was achieved by combining Google Scholar and the Jensen-Shannon measure. The algorithm stops to iterate when no more improvement can be obtained, in this case after 4 iterations, resulting in a final MAP of .619.

Although α-Skew outperforms the simpler measures in ranking nouns, its performance on verbs is worse than the performance of Weighted Lin. While Lee (1999) argues that α-Skew's asymmetry can be advantageous for nouns, this probably does not hold for verbs: verb hierarchies have much shallower structure than noun hierarchies with most verbs concentrated on one level (Miller et al., 1990). This would explain why JS, which is symmetric compared to the α-Skew metric, performed better in our experiments.

In the evaluation presented here we therefore use Google Scholar data and the JS measure. An additional improvement (MAP=.630) is achieved when we incorporate a filter based on the following hypothesis: goal-type verbs should be more likely to have their direct objects preceded by indefinite articles rather than definite articles or possessive determiners (because a new method is introduced) whereas continuation-type verbs should prefer definite articles with their direct objects (as an existing method is involved).

3 Syntactic variants and semantic filters

The syntactic variant extractor takes as its input the raw text and the lists of verbs and nouns generated by the lexical bootstrapper. After RASP-parsing the input text, all instances of the input verbs are located and, based on the grammatical relations output by RASP[6], a set of relevant entities and modifiers for each verb is constructed, grouped into five categories (cf. Fig. 6).

The agent of the verb (e.g., "*We* adopt... ...adopted by *the author*"), the agent's determiner and related adjectives.
The direct object of the verb, the object's determiner and adjectives, in addition to any post-modifiers (e.g., "...apply a method *proposed by [1]* ..." , "...follow an approach *of [1]* ..."
Auxiliaries of the verb (e.g., "In a similar manner, we *may* propose a ...")
Adverbial modification of the verb (e.g., "We have *previously* presented a ....")
Prepositional phrases related to the verb (e.g., "*In this paper* we present...", "...adopted *from their work*")

Figure 6: Grammatical relations considered

Next, semantic filters are applied to each of the potential candidates (represented by the extracted entities and modifiers), and a fitness score is calculated. These constraints encode semantic principles that will apply to all cue phrases of that rhetorical category. Examples for constraints are: if work is referred to as being done in previous own work, it is probably not a goal statement; the work in a goal statement must be presented *here* or *in the current paper* (the concept of 'here-ness"); and the agents of a goal statement have to be the authors, not other people. While these filters are manually defined, they are modular, encode general principles, and can be combined to express a wide range of rhetorical contexts. We verified that around 20 semantic constraints are enough to cover a large sets of different cue phrases (the 1700 cue phrases from Teufel (1999)), though not all of these are implemented yet.

A nice side-effect of our approach is the simple characterisation of a cue phrase (by a syntactic relationship, some seed words for each concept, and some general, reusable semantic constraints). This characterisation is more informative and specific than string-based approaches, yet it has the potential for generalisation (useful if the cue phrases are ever manually assessed and put into a lexicon).

Fig. 7 shows successful extraction examples from our corpus[7], illustrating the difficulty of the task: the system correctly identified sentences with syntactically complex goal-type and continuation-type cue phrases, and correctly rejected deceptive variants[8].

[6]The grammatical relations used are nsubj, dobj, iobj, aux, argmod, detmod, ncmod and mod.

[7]Numbers after examples give CmpLg archive numbers, followed by sentence numbers according to our preprocessing.

[8]The seeds in this example were [*analyse, present*] & [*architecture, method*] (for goal) and [*improve, adopt*] & [*model, method*] (for continuation).

Correctly found:
Goal-type:

*What we aim in this paper is to **propose** a **paradigm** that enables partial/local generation through decompositions and reorganizations of tentative local structures.* (9411021, S-5)

Continuation-type:

*In this paper we have discussed how the lexicographical **concept** of lexical functions, introduced by Melcuk to describe collocations, can be **used** as an interlingual device in the machine translation of such structures.* (9410009, S-126)

Correctly rejected:
Goal-type:

*Perhaps the **method proposed** by Pereira et al. (1993) is the most relevant in our context.* (9605014, S-76)

Continuation-type:

*Neither Kamp nor Kehler **extend** their copying/ substitution **mechanism** to anything besides pronouns, as we have done.* (9502014, S-174)

Figure 7: Sentences correctly processed by our system

4 Gold standard evaluation

We evaluated the quality of the extracted phrases in two ways: by comparing our system output to gold standard annotation, and by human judgement of the quality of the returned sentences. In both cases bootstrapping was done using the seed tuples [*analyse*, *present*] & [*architecture*, *method*]. For the gold standard-evaluation, we ran our system on a test set of 121 scientific articles drawn from the CmpLg corpus (Teufel, 1999) – entirely different texts from the ones the system was trained on. Documents were manually annotated by the second author for (possibly more than one) goal-type sentence; annotation of that type has been previously shown to be reliable at K=.71 (Teufel, 1999). Our evaluation recorded how often the system's highest-ranked candidate was indeed a goal-type sentence; as this is a precision-critical task, we do not measure recall here.

We compared our system against our reimplementation of Ravichandran and Hovy's (2002) paraphrase learning. The seed words were of the form {goal-verb, goal-noun}, and we submitted each of the 4 combinations of the seed pair to Google Scholar. From the top 1000 documents for each query, we harvested 3965 sentences containing both the goal-verb and the goal-noun. By considering all possible substrings, an extensive list of candidate patterns was assembled. Patterns with single occurrences were discarded, leaving a list of 5580 patterns (examples in Fig. 8). In order to rank the patterns by precision, the goal-verbs were submitted as queries and the top 1000 documents were downloaded for each. From these,

we <verb> a <noun> for
of a new <noun> to <verb> the
In this section , we <verb> the <noun> of
the <noun> <verb> in this paper
is to <verb> the <noun> after

Figure 8: Examples of patterns extracted using Ravichandran and Hovy's (2002) method

Method	Correct sentences
Our system with bootstrapping	88 (73%)
Ravichandran and Hovy (2002)	58 (48%)
Our system, no bootstrapping, WordNet	50 (41%)
Our system, no bootstrapping, seeds only	37 (30%)

Figure 9: Gold standard evaluation: results

the precision of each pattern was calculated by dividing the number of strings matching the pattern instantiated with both the goal-verb and all WordNet synonyms of the goal-noun, by the number of strings matching the patterns instantiated with the goal-verb only. An important point here is that while the tight semantic coupling between the question and answer terms in the original method accurately identifies all the positive and negative examples, we can only approximate this by using a sensible synonym set for the seed goal-nouns. For each document in the test set, the sentence containing the pattern with the highest precision (if any) was extracted as the goal sentence.

We also compared our system to two baselines. We replaced the lists obtained from the lexical bootstrapping module with a) just the seed pair and b) the seed pair and all the WordNet synonyms of the components of the seed pair[9].

The results of these experiments are given in Fig. 9. All differences are statistically significant with the χ^2 test at p=.01 (except those between Ravichandran/Hovy and our non-bootstrapping/WordNet system). Our bootstrapping system outperforms the Ravichandran and Hovy algorithm by 34%. This is not surprising, because this algorithm was not designed to perform well in tasks where there is no clear negative context. The results also show that bootstrapping outperforms a general thesaurus such as WordNet.

Out of the 33 articles where our system's favourite was not an annotated goal-type sentence, only 15 are due to bootstrapping errors (i.e., to an incorrect ranking of the lexical variants), corre-

[9]Bootstrapping should in principle do better than a thesaurus, as some of our correctly identified variants are not true synonyms (e.g., *theory* vs. *method*), and as noise through overgeneration of unrelated senses might occur unless automatic word sense diambiguation is performed.

System chose:	but should have chosen:
derive set	compare model
illustrate algorithm	present formalisation
discuss measures	present variations describe modifications propose measures
accommodate material	describe approach
examine material	present study

Figure 10: Wrong bootstrapping decisions

	Ceiling	System	Baseline
Exp. A	3.91	3.08	1.58
Exp.B	4.33	3.67	2.50

Figure 11: Extrinsic evaluation: judges' scores

sponding to a 88% accuracy of the bootstrapping module. Examples from those 15 error cases are given in Fig. 10. The other errors were due to the cue phrase not being a transitive verb–direct object pattern (e.g. *we show that, our goal is* and *we focus on*), so the system could not have found anything (11 cases, or an 80% accuracy), ungrammatical English or syntactic construction too complex, resulting in a lack of RASP detection of the crucial grammatical relation (2) and failure of the semantic filter to catch non-goal contexts (5).

5 Human evaluation

We next perform two human experiments to indirectly evaluate the quality of the automatically generated cue phrase variants. Given an abstract of an article and a sentence extracted from the article, judges are asked to assign a score ranging from 1 (low) to 5 (high) depending on how well the sentence expresses the goal of that article (Exp. A), or the continuation of previous work (Exp. B).

Each experiment involves 24 articles drawn randomly from a subset of 80 articles in the CmpLg corpus that contain manual annotation for goal-type and continuation-type sentences. The experiments use three external judges (graduate students in computational linguistics), and a Latin Square experimental design with three conditions: Baseline (see below), System-generated and Ceiling (extracted from the gold standard annotation used in Teufel (1999)). Judges were not told how the sentences were generated, and no judge saw an item in more than one condition.

The baseline for Experiment A was a random selection of sentences with the highest $TF*IDF$ scores, because goal-type sentences typically contain many content-words. The baseline for experiment B (continuation-type) were randomly selected sentences containing citations, because they often co-occur with statements of continuation. In both cases, the length of the baseline sentence was controlled for by the average lengths of the gold standard and the system-extracted sentences in the document.

Fig. 11 shows that judges gave an average score of 3.08 to system-extracted sentences in Exp. A, compared with a baseline of 1.58 and a ceiling of 3.91[10]; in Exp. B, the system scored 3.67, with a higher baseline of 2.50 and a ceiling of 4.33. According to the Wilcoxon signed-ranks test at $\alpha = .01$, the system is indistinguishable from the gold standard, but significantly different from the baseline, in both experiments. Although this study is on a small scale, it indicates that humans judged sentences obtained with our method as almost equally characteristic of their rhetorical function as human-chosen sentences, and much better than non-trivial baselines.

6 Conclusion

In this paper we have investigated the automatic acquisition of semi-fixed cue phrases as a bootstrapping task which requires very little manual input for each cue phrase and yet generalises to a wide range of syntactic and lexical variants in running text. Our system takes a few seeds of the type of cue phrase as input, and bootstraps lexical variants from a large corpus. It filters out many semantically invalid contexts, and finds cue phrases in various syntactic variants. The system achieved 80% precision of goal-type phrases of the targeted syntactic shape (88% if only the bootstrapping module is evaluated), and good quality ratings from human judges. We found Google Scholar to perform better than BNC as source for finding hypotheses for lexical variants, which may be due to the larger amount of data available to Google Scholar. This seems to outweigh the disadvantage of only being able to use POS patterns with Google Scholar, as opposed to robust parsing with the BNC.

In the experiments reported, we bootstrap only from one type of cue phrase (transitive verbs and direct objects). This type covers a large proportion of the cue phrases needed practically, but our algorithm should in principle work for any kind of semi-fixed cue phrase, as long as they have two core concepts and a syntactic and semantic

[10]This score seems somewhat low, considering that these were the best sentences available as goal descriptions, according to the gold standard.

CUE PHRASE: "(previous) **methods fail**" (Subj–Verb)
VARIANTS SEED 1: methodology, approach, technique...
VARIANTS SEED 2: be cursed, be incapable of, be restricted to, be troubled, degrade, fall prey to, ...
CUE PHRASE: "**advantage** over **previous** methods" (NP–PP postmod + adj–noun premod.)
VARIANTS SEED 1: benefit, breakthrough, edge, improvement, innovation, success, triumph...
VARIANTS SEED 2: available, better-known, cited, classic, common, conventional, current, customary, established, existing, extant,...

Figure 12: Cues with other syntactic relationships

relation between them. Examples for such other types of phrases are given in Fig. 12; the second cue phrase involves a complex syntactic relationship between the two seeds (or possibly it could be considered as a cue phrase with three seeds). We will next investigate if the positive results presented here can be maintained for other syntactic contexts and for cue phrases with more than two seeds.

The syntactic variant extractor could be enhanced in various ways, eg. by resolving anaphora in cue phrases. A more sophisticated model of syntactically weighted vector space (Pado and Lapata, 2003) may help improve the lexical acquisition phase. Another line for future work is bootstrapping meaning across cue phrases within the same rhetorical class, e.g. to learn that *we propose a method for X* and *we aim to do X* are equivalent. As some papers will contain both variants of the cue phrase, with very similar material (*X*) in the vicinity, they could be used as starting point for experiments to validate cue phrase equivalence.

7 Acknowledgements

This work was funded by the EPSRC projects CITRAZ (GR/S27832/01, "Rhetorical Citation Maps and Domain-independent Argumentative Zoning") and SCIBORG (EP/C010035/1, "Extracting the Science from Scientific Publications").

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*.

Regina Barzilay and Lillian Lee. 2002. Bootstrapping lexical choice via multiple-sequence alignment. In *Proc. of EMNLP*.

Ted Briscoe and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proc. of LREC*.

Lou Burnard, 1995. *Users Reference Guide, British National Corpus Version 1.0*. Oxford University, UK.

John Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of Linguistically Interpreted Corpora (LINC-99), EACL-workshop*.

Ute Essen and Volker Steinbiss. 1992. Co-occurrence smoothing for stochastic language modelling. In *Proc. of ICASSP*.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proc. of the ACL*.

Edvard Hovy and Chin-Yew Lin. 1998. Automated text summarization and the Summarist system. In *Proc. of the TIPSTER Text Program*.

Ken Hyland. 1998. Persuasion and context: The pragmatics of academic metadiscourse. *Journal of Pragmatics*, 30(4):437–455.

Christian Jacquemin, Judith Klavans, and Evelyn Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proc. of the ACL*.

Julian Kupiec, Jan O. Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proc. of SIGIR-95*.

Lillian Lee. 1999. Measures of distributional similarity. In *Proc. of the ACL*.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE transactions on Information Theory*, 37(1):145–151.

Frederique Lisacek, Christine Chichester, Aaron Kaplan, and Sandor Agnes. 2005. Discovering paradigm shift patterns in biomedical abstracts: Application to neurodegenerative diseases. In *Proc. of the SMBM*.

George Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Five papers on WordNet. Technical report, Cognitive Science Laboratory, Princeton University.

Greg Myers. 1992. In this paper we report...—speech acts and scientific facts. *Journal of Pragmatics*, 17(4):295–313.

Sebastian Pado and Mirella Lapata. 2003. Constructing semantic space models from parsed corpora. In *Proc. of ACL*.

Chris D. Paice. 1981. The automatic generation of literary abstracts: an approach based on the identification of self-indicating phrases. In Robert Norman Oddy, Stephen E. Robertson, Cornelis Joost van Rijsbergen, and P. W. Williams, editors, *Information Retrieval Research*, Butterworth, London, UK.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proc. of the ACL*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proc. of the ACL*.

Ellen Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proc. of AAAI-93*.

Simone Teufel. 1998. Meta-discourse markers and problem-structuring in scientific articles. In *Proceedings of the ACL-98 Workshop on Discourse Structure and Discourse Markers*.

Simone Teufel. 1999. *Argumentative Zoning: Information Extraction from Scientific Text*. Ph.D. thesis, School of Cognitive Science, University of Edinburgh, UK.

Cornelis Joost van Rijsbergen. 1979. *Information Retrieval*. Butterworth, London, UK, 2nd edition.

Hua Wu and Ming Zhou. 2003. Synonymous collocation extraction using translation information. In *Proc. of the ACL*.