Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference

Multiword Term Extraction through Lexical Head Selection

Dirk De Hertog, Piet Desmet
KU Leuven, imec
Etienne Sabbelaan 53
8500 Kortrijk

Abstract

We propose a semantically inspired Multiword Term Extractor that selects candidates for which the headword belongs to a seed list of approved single word terms. In order to achieve this without resorting to the computational complexities of a full parser, we apply a selection pipeline that leverages lightweight NLP-tools such as POS-taggers, chunkers and a self-devised head detection module.

Introduction

Automatic Term Extraction (ATE) is a subfield of Information Extraction that aims to extract semantically interesting terms for a specific domain. For some domains, terms resort under the name of certain topic classes. For instance, in the human resource domain interesting terms might be categorized as 'skills' or 'job titles'. Specifically for the detection and extraction of MultiWord Term (MWT) candidates, current approaches in ATE are based on syntactic and/or statistical information derived from specialized corpora.

While the latter procedures certainly prove useful to rank term candidates, we would like to present a complementary and more semantically inspired approach that results in a more finalized selection. The procedure we propose selects MWT candidates for which the headword belongs to a seed list of approved single word terms. In order to achieve this without being confronted to the computational runtime of applying a full syntactic parser to large corpora, we will present a selection pipeline that leverages lightweight NLP-tools such as POS-taggers, chunkers and a self-devised head detection module. The main contributions include the creation of a specialized head-tagger using existing resources and its application to the field of ATE.

The remainder of this paper is structured as follows. Section 2 discusses current extraction and selection methods. Section 3 elaborates on the working procedure of the head selection pipeline. This includes a detailed step-by-step overview of how we proceeded with the training of the head detection module. Section 4 presents the results and discusses them. A brief summary wraps up and points towards future loci of investigation.

Previous approaches

Linguistic approaches in ATE benefit from the particular appearance of MWTs as noun phrases (NPs). In theory, terms cover a wider range of expressions than NPs, but in practice, it is often the only type of phrase targeted by the extraction process. Success of the extraction process is measured at the hand of precision and recall scores by comparing a manually annotated gold standard to a (ranked) list of candidates.

Early approaches use an NP's relatively fixed morphosyntactic structure to engineer so called linguistic filters, morphosyntactic templates that are matched to POS-tagged sentences in order to extract positive matches. Linguistic templates take an arbitrary number of words as possible candidates. Justeson and Katz (1995) and Daille, Gaussier, and Langé (1994) use such methods for respectively English and French. A more recent paradigm, closely related to linguistic filters, uses chunkers (Wermter and Hahn 2005) and parsers to extract NPs from a given collection of texts. They are computationally more expensive, yet arguably more performant. Linguistic approaches are seldom used without any additional selection criteria.

Statistical criteria (Smadja 1993) rank the candidates according to their co-occurrence. Advanced measures are mostly used to identify bigrams. As a consequence, early linguistic approaches likewise often limit themselves to bigrams. Longer term candidates are often sparse and mere frequency is already extremely informative, which leads some researchers to claim that "you can't beat frequency" (Wermter and Hahn 2006).

It is also possible to use statistical information by itself without any further linguistic filters (Pantel and Lin 2001). An interesting line of research within ATE is the development of techniques that rely on the comparison of frequency signatures of n-grams to establish independence of a given chunk. C-NC value (Frantzi, Ananiadou, and Mima 2000) is a measure almost exclusively used within ATE, but similar approaches such as the localMax algorithm (Da Silva et al. 1999) have found more widespread usage.

Earlier approaches investigated how to manually combine linguistic and statistical information. A rather recent development steps away from such manual selection procedures and resorts to supervised machine learning techniques in order to discriminate and select interesting term candidates (Turney 2000; Foo and Merkel 2010).

In addition to statistics, the precision of the extraction process is boosted by filtering out a list of stopwords that are unlikely to constitute terms. The lists typically contain general language words such as 'thing', 'several', 'previous'. A frequency list of words is often used to facilitate their identification.

Limitations The current selection procedures share some limitations in the results they offer, which could briefly be described as prefinal. We point to two related reasons for this: the lack of a semantic component and hence the lack of an objective criterion that decides on term inclusion.

The precision of the ranked lists depends highly on the quality of relevant text selection for the domain (corpus-compilation), rather than the quality of the statistics used for ranking. Furthermore, the use of ranked vocabulary lists can be impractical. As it stands, current ranking procedures take a list of MWT-candidates and output the same reranked list. They contain a high number of candidates for which an empirically chosen cut-off value often determines whether they are included in a final candidate list presented to an expert for manual validation. Also, the ranking procedures make it difficult to include low-frequency occurrences. The latter, while potentially interesting from a semantic point of view, unfortunately get downranked through the use of global frequency statistics or can not be included in an analysis that relies on more advanced methods, because they lack statistical power.

Ideally, by leveraging more meaningful and better structured information, provided by the selection procedure we propose, the presented results would remedy both concerns.

Proposed selection procedure

We propose a selection procedure that is both linguistically inspired and computationally attractive, which utilizes information concerning the NP's syntactic head (see Nakagawa and Mori (2002) for a similar approach applied to compound nouns). It is meant as an addition to current selection procedures and relies on the presence or creation of an approved list of single word terms. It is different from previous approaches as it relies on an objective head selection criterion, instead of a statistic reranking of candidates, and thus presents a finalized candidate list. Figure 1 visually presents the proposed selection pipeline.

A targeted selection of MWT candidates using lexical resources in combination with syntactic head information can be defended based on linguistic grounds (Lieber 2005; Pollard and Sag 1994). All NPs with compositional semantics adhere to the following set of rules.

1. The head noun acts both as a semantic and a syntactic head within the full NP.
2. A more specified NP has a direct lexical relationship with the simple head noun, namely that of a subordinate with a type-of relationship.
3. NPs that share a head noun are co-hyponyms.

Because of the existence of non-compositional word combinations (e.g. 'cold start'), one could argue that it is dangerous to assume compositional semantics. However, we point to the high number of newly created word combinations in technical texts as an argument in favor of the assumption of compositionality. The argument might be more clear when presented in terms of intelligibility. In order to maintain a high level of understanding among readers/experts, they necessarily rely on compositional, or endocentric, combinatorial semantics. By relying on head-driven phrase formation (e.g. 'inventory of materials') and head-driven compounding (e.g. 'center assistant planner'), new terms can be introduced easily.

In order to provide the needed structure to identify heads, we look at the current NLP options: templatic detection of NPs, chunkers, and full syntactic parsers. The use of chunkers and parsers in the MWT detection process is appealing because they impose certain desirable characteristics on the extracted candidates. The most important one being that NPs come in the form of complete chunks for which a language user can resolve the reference. Note that purely frequency-based techniques seem like an interesting choice from a computational point of view. However, in their current state, they are largely unsuccessful at selecting such informationally interesting full NPs.

Templatic detection and chunkers require manual rule-based engineering to determine the usual position of a template's syntactic head. Such rules often limit themselves to constructions that involve attributive adjectives (e.g. '*technical* communications') and specifying nouns (e.g. '*production* drawing') , but can be extended to include prepositional attachments (e.g. 'auditor with dwp knowledge') as well. Full syntactic parsers provide the necessary phrase structure or dependencies to determine the syntactic head. However, the computational cost involved in running full parsers on large-scale corpora often render it an unattractive choice.

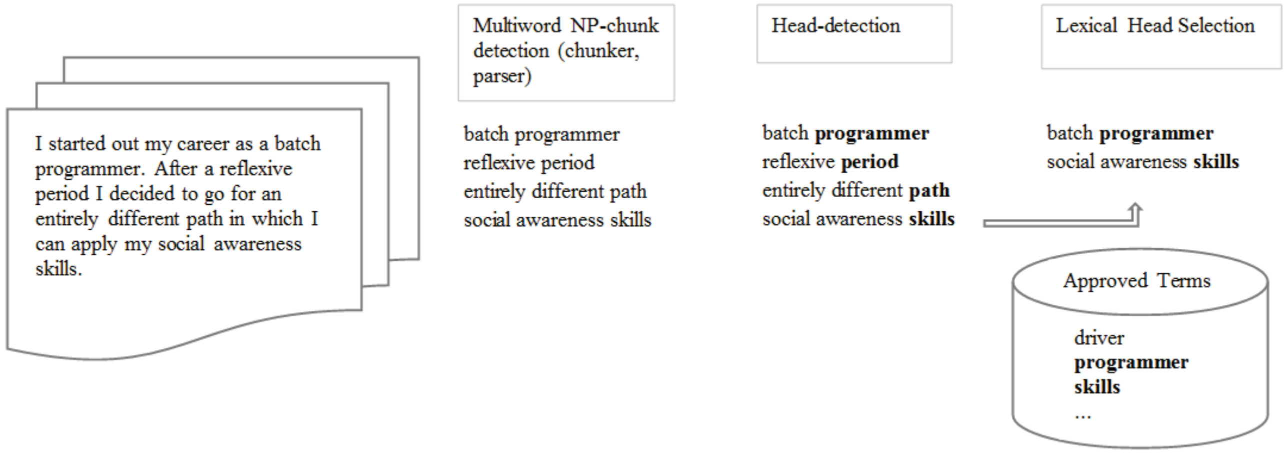
We propose an alternative procedure that leverages the output of a dependency parser to automatically create a training set, and then use a supervised learning approach to decide on the head position of an NP. If that head position is then filled with a known term, such as 'manager' or 'engineer', we include it in a list of extracted terms. Note that English serves as an example in this specific study but that the proposed procedure is intended to be generally applicable without in-depth knowledge of the language it is applied to.

Methodology

Materials We have at our disposal a large collection of English cover letters from which we sampled 10,000 documents. The document collection has been provided by an industrial partner, active in the human resource domain, participating in the project in which this research can be situated.

We use the command line tool of the Stanford CoreNLP parser (Klein and Manning 2003) to pre-process (tokenize, POS-tag) the material and create dependency parses and phrase structure. We explicitly chose not to lowercase the material due to the high amount of uppercase abbreviations and proper nouns present in the texts. NLTK (Bird, Klein, and Loper 2009) is used to extract chunks (subtrees) from

Figure 1: Proposed selection pipeline



the detected phrase structure. We filter out tagged-material that contains conjunctions (e.g. ’community pharmacy and healthcare’) and punctuation marks in order to impose the constraint that each NP is headed by a single noun. By iterating over all subtrees, smaller interesting NP chunks do not get lost in the process. This procedure yields 34,154 phrases (i.e. tokens), belonging to 7,112 different lexical combinations (i.e. types), of varying levels of complexity and length. At this point we proceed with the creation of a training set for the head detection module (infra).

The same NP-candidates are reused in the selection process based on a seed list. The seed set consists of 1,249 manually selected[1] terms. All terms are considered to be skills or jobs. All terms are lower-cased in order to match the lower-cased NP-candidate list in the best possible way.

Creating a head-training set Dependency parsers provide the detailed information we require to determine which word is the head noun within selected NPs[2]. We use that ability to automatically create annotated resources for a supervised learning approach that specializes in the detection of heads.

Dependency parsers provide structure to a sentence and show how words are related. For our purposes we require it to establish which word is the head of a given chunk. The head of a phrase has as a desirable property that it is governed by an element external to the phrase. Furthermore, we attempt to select only those phrases with a single head. Therefore, if we know the phrase boundaries through a chunking module, and also have a dependency parse, we can uniquely identify the phrasal heads. If the head is a noun, then the phrase, by definition, will also be a noun.

Chunks that are not headed by a noun are discarded. Instances starting with a determiner or a pronoun are truncated. By doing so we create a training sample of 325,797 phrases with positional information about their syntactic heads.

Table 1: Performance of Head-model

	Precision	**Recall**	**F-score**	**Support**
Dependent	0.98	0.98	0.98	67157
Head	0.96	0.96	0.96	32447
Avg	0.98	0.98	0.98	99604

Training a head-model We train an averaged perceptron learner in scikit-learn (Pedregosa et al. 2011) where we consider each NP as a word sequence with corresponding tags that identify either H(ead) or D(ependent).The model is trained for English on 90% of the tagged material, and tested on the remaining 10%. The standard settings for the scikit-learn perceptron are maintained; five iterations are deemed sufficient due to the high number of examples and repitition in the training set. The model is based purely on POS-information. The training features include a window of four words preceding and following the target word. We include POS-combinations for all lengths up until 3 within the same context window. Table 1 shows the performance of the model.

We motivate the use of a supervised learning technique as follows. Wrong tags can be introduced at each stage of the NLP-processing; the head tagging, the POS-tagging and the chunking. However, given numerous different contexts, used as features for the predictive NLP-models, we hypothesize that the majority of the POS-tagging and consequently, the chunking and the heading, proceeds correctly. We assume the model will take this information and generalize correctly, even if we present it with a minority of wrong tags.

A generalization to other languages or language families would require some extensions. The exclusive use of struc-

[1] By human resource specialists from the industrial partner.

[2] This requires the NPs to exclude conjunctions, because of the need of a single head per phrase

tured information implies the assumption that lexical items do not exert any influence over head-positions. This holds true for English, in which the compounding system follows the right hand rule (Williams 1981), stipulating that the head of a compound word (in Germanic languages) is situated at the right-hand side. For some languages however, the assumption of a fixed head-position (either left or right) is untrue. For instance, in Romance languages, the head of a compound can occur in either position. As a side note, this divide can be attributed to an 'old' and a 'new' compounding system, where the old system goes back to its Latin roots. In order to train a model specifically for such languages we would necessarily have to add lexical features to the model.

Applying the head-model The trained model can be applied in two different ways; *ad-hoc*, when a single instance is provided as input, or in batch mode, when a collection of texts is provided.

The head-tagger determines a head position for a POS-tagged NP-chunk. The manner in which these NP-candidates have been identified is of minor importance and depends mostly on the user's preferred way of identification. Given the certainty that each POS-sequence has a uniquely defined deterministic head, we can apply the head-tagger to a single POS-sequence and use the found position to each of the lexical instances that are tagged with that pattern. This results in a fairly limited number of required head detections, which makes it extremely efficient.

Once the structural head position for each example is known, we can identify its lexical head. The head nouns are subsequently matched to the known skill and job-related terms. Table 2 exemplifies the procedure and presents the head words in bold when they match approved terms.

Results

We subject 7,112 different lexical NP-chunks to the selection procedure, i.e. a single token representation for each of the n-grams we extracted, corresponding to a single POS-sequence. We will describe and analyse the results from different perspectives. First the head tagger is investigated to see how it performs on the data. Second we investigate precision scores of the extracted candidates. An error analysis then investigates future loci of research and improvement.

Head tagger The training phase of the head tagger does not impose the constraint that any sequence has exactly one head. The discriminative learning approach takes into account four context-words in any direction, but the target in itself is either the tag Head or Dependent. Interestingly enough, there are no occurrences of phrases that are tagged with more than one head. However, out of the 7,112 candidates, 2,741 are tagged as headless.

A (sampled) more detailed analysis uncovers that many 'headless' sequences contain tagging mistakes, are wrongly identified as an NP-chunk or are in fact incomplete chunks. For short sequences, it is mostly POS mistakes that cause the headless tagging. In this respect, not lowercasing the corpus before parsing it, seems to have the undesired effect that capitalised common nouns are often tagged as proper nouns. Spelling mistakes and the inclusion of non-existing words also has an effect on the quality of the tags. For longer sequences, similar mistakes can be identified. The wrong POS-tags inevitably also influence the quality of the chunker, which at times has problems establishing the correct boundaries of the chunk altogether.

These findings thus lead to some interesting behavior of the head detection algorithm and the manner in which the learner generalized over the high amount of low-quality input data. The tagger has a positive effect as it can single out mistakes introduced by either the POS-tagger or the chunker. As such, we see applications where head-tagging could help in post-processing of tagged and chunked texts. As a small experiment we selected some highly frequent n-grams that had been tagged in different ways and investigated the effect on the head tagging. The correct POS-sequence was indeed tagged with a head position, while the others were left headless.

However, there are also longer sequences that are consistently not recognised by the tagger; this includes cases with complex prepositional attachments involving verbs and participles, such as 'availability before releasing the production orders'.

Precision and error analysis We sampled 500 candidates (from the total of 4,371 tagged chunks) and scored them for precision. A single annotator decided whether the detected candidates fit the description of a 'job title' or a 'job skill'. 421 (83.6 %) proved to be valid MWT candidates, which is considered by the authors as a successful start towards a highly accurate selection procedure.

We provide a detailed overview of the types of errors we encountered and grouped them in three categories: tool-related (Table 3), resource-related (Table 4) and semantic reasons (Table 5).

It becomes clear that the majority of errors can be traced back to the NLP-tools used in the process. Further analysis shows that incorrect chunks that are selected at this point, i.e. which have not been automatically removed because the head-tagger was unable to identify a head, are mostly (15 out of 24) constructions that end in a possesive 's'. Within the mistakes introduced by NLP-tools, the high precision of the head tagger, constituting a mere 3 mistakes, is noteworthy. Admittedly, this number provides an optimistic view at this point, considering the compounded filtering effect of the lexical selection step and the fact that numerous cases are left headless.

Resource-related errors include head words deemed irrelevant for the domain (e.g. 'sea', 'rest'), and written material we categorized as language errors. For instance, while we understand the value of the skill 'down loading' and appreciate the rather inventive 'responsibility freelance copywriter', we did not accept them as relevant for the task at hand. A number of mistakes can also be attributed to the appearance of repeated enumeration without the use of conjunctions or punctuation marks (e.g. 'temporary work

Table 2: Results of Selection Pipeline

N-gram	NP POS-sequence	Head-sequence	Head-word
departure control	NN NN	D H	**control**
implementation of a new robotic system	NN IN DT JJ JJ NN	H D D D D D	**implementation**
annual sales goals	JJ NNS NNS	D D H	goals
large regional shopping centres	JJ JJ NN NNS	D D D H	centres
excellent customer service skills	JJ NN NN NNS	D D D H	**skills**
Maintenance Technician Role			role
part time basis			basis
case management system	NN NN NN	D D H	**system**
passenger security screening			**screening**
travel agency customer			customer
short term			term
photographic assistant			**assistent**
correct signatory	JJ NN	D H	signatory
timely manner			manner
final audit			**audit**

Table 3: Errors Related to NLP-tools

Reason	Count
Chunk is not an NP	24
Wrong Pos-tag	5
Chunk is a Proper Name	4
Item is a full sentence	3
Wrong element is tagged as head	3
Chunk has missing end slot	2

Table 4: Errors Related to Available Resources

Reason	Count
Undetectable enumeration	15
Language errors	10
Wrong element in seedlist	5

Table 5: Errors Related to Semantics

Reason	Count
Uninteresting attributive adjective	6
Uninteresting prepositional attachment	2
Uninteresting specifier	1

company equipment supervision team').

Notice the select number of mistakes due to semantically uninteresting combinations. In fact, we could only identify 4 adjectives ('young', 'defined', 'new', 'full'), a single specifier ('start') and 2 uninteresting prepositional attachments ('as required', 'by myself'). For cv's, a text genre which can be described as highly informative and relatively formal, it follows expectations that everyday general words only occur sporadically. The truncation of functional words (such as determiners and pronouns) in a chunk's starting position seems effective for filtering out unwanted word combinations. Still, the low number of mistakes also confirms the data we have at our disposal is of high quality. Longer expressions in general become more difficult to assess relevancy for. While the productive compounding system can yield rather long candidates (e.g. leading provider of cloud based shared service solutions), we have the intuition that there is not much terminological interest in expressions that have more than one or two prepositional attachments (e.g. senior member of steering group company representation on cruise liners), or that have as a complement a full subordinate clause (e.g. 'department manager to help maintain the high standard that IBM required of the customer service team'). The latter is in fact a case for the use of simple syntactic templates to identify relevant NPs.

Summary

This paper explores the use of a specialized head-tagging module for NPs and applies it to the task of automatic MWT-extraction. The selection procedure is linguistically inspired as it exploits the knowledge that nominal heads are informative with regard to the compound or phrase they belong to. It is meant to supplement current selection procedures and relies on the presence or creation of an approved list of single word terms. It is also computationally attractive as the specialized head detection module is a lightweight addition to existing NLP-tools required to provide structure to language use.

As far as the MWT extraction task is concerned, a sampled verification of the results show high precision (83.6 %). Most errors can be traced back to the use of the other NLP-tools, namely the POS-tagger and the chunker. Remarkably few mistakes can be attributed to semantic reasons, testifying to the validity of the selection procedure as presented, in combination with yet this achievement is also undoubtedly related to the high quality of text material we have at our disposal.

What concerns the head detection module, we see performance reaching high levels during the test phase: precision, recall and F-score have an average of 98%. Applying the model to NP chunks shows however that the head detection module leaves many instances headless. We identified wrong tags, introduced by the POS-tagger, and wrong demarcations, introduced by the chunker, as possible reasons for this behavior. However, we also notice that for select cases, the head tagger makes a wrong (structural) assessment. We leave it for future work to investigate how we can improve the results for the latter group and exploit the behavior of the former. The mixed results in any case suggest a further need to fine-tune the configuration of the supervised learner setup and experiment with different feature settings, one for instance that includes information about previous head tags, and whether a head has been found for the construction as a whole.

Acknowledgements

ASIMOV is a project realized in collaboration with imec. Project partners are Actonomy, Crosslang, de Persgroep and IDLab (uGent), with project support from VLAIO (Flanders Innovation & Entrepreneurship).

References

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Da Silva, J.; Dias, G.; Guilloré, S.; and Pereira Lopes, J. 1999. Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. *Progress in Artificial Intelligence* 849–849.

Daille, B.; Gaussier, É.; and Langé, J.-M. 1994. Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th International Conference on Computational Linguistics COLING94*, volume 1, 515–521. Association for Computational Linguistics.

Foo, J., and Merkel, M. 2010. Using machine learning to perform automatic term recognition. In *Proc of the 7th LREC - Wksp on Methods for automatic acquisition of Language Resources and their Evaluation Methods*, 49–54.

Frantzi, K.; Ananiadou, S.; and Mima, H. 2000. Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries* 3:115–130.

Justeson, J. S., and Katz, S. M. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering* 1(01):9–27.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, volume 1, 423–430.

Lieber, R. 2005. *English Word-Formation Processes*. Dordrecht: Springer Netherlands. 375–427.

Nakagawa, H., and Mori, T. 2002. A simple but powerful automatic term extraction method. *COLING02 on COMPUTERM 2002 second international workshop on computational terminology* 14:1–7.

Pantel, P., and Lin, D. 2001. A Statistical Corpus-Based Term Extractor. In *Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, AI '01, 36–46. London, UK, UK: Springer-Verlag.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research* 12:2825–2830.

Pollard, C., and Sag, I. A. 1994. Head-driven phrase structure grammar.

Smadja, F. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics* 19:143–177.

Turney, P. D. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval* 2(4):303–336.

Wermter, J., and Hahn, U. 2005. Finding new terminology in very large corpora. *Proceedings of the 3rd international conference on* 137–144.

Wermter, J., and Hahn, U. 2006. You Can't Beat Frequency (Unless You Use Linguistic Knowledge) – A Qualitative Evaluation of Association Measures for Collocation and Term Extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, number July in ACL '06, 785–792. Association for Computational Linguistics.

Williams, E. 1981. On the Notions "Lexically Related" and "Head of a Word". *Linguistic Inquiry* 12(2):245–274.