Extracting and matching patent in-text references to scientific publications

Suzan Verberne[0000−0002−9609−9505], Ioannis Chios, and Jian Wang[0000−0003−0520−737X]

Leiden Institute of Advanced Computer Science,
Leiden University, Leiden, The Netherlands
s.verberne@liacs.leidenuniv.nl

Abstract. References in patent texts to scientific publications are valuable for studying the links between science and technology but are difficult to extract. This paper tackles this challenge, specifically, we extract references embedded in USPTO patent full texts and match them to Web of Science (WoS) publications. We approach the reference extraction problem as a sequence labelling task, training CRF and Flair models. We then match references to the WoS using regular expression patterns. We train and evaluate the reference extraction models using cross validation on a sample of 22 patents with 1,952 manually annotated in-text references. Then we apply the models to a large collection of 33,338 biotech patents. We find that CRF obtains better results on citation extraction than Flair, with precision scores of around 90% and recall of around 85%. However, Flair extracts much more references from the large collection than CRF, and more of those can be matched to WoS publications. We find that 88% of the extracted in-text references are not listed on patent front page, suggesting distinct roles played by in-text and front-page references. CRF and Flair collectively extract 603,457 references to WoS publications that are not listed on the front page. In addition to the 1.17 Million front-page references in the collection, this is a 51% increase in identified patent–publication links compared with only relying on front-page references.

Keywords: citations · patents · sequence labelling

1 Introduction

Scientific non-patent references (sNPRs, i.e., references in patents to scientific literature) provide a paper trail of the knowledge flow from science to technological innovation. They have wide applications for science and innovation studies, science policy, and innovation strategy [15, 8, 12, 6, 1, 19, 21]. However, the current practice relies exclusively on patent *front-page references* but neglects the more difficult patent *in-text references*. Front-page references are the references listed on the front page of the patent document, which are deemed as relevant prior art for assessing the patentability by inventors, patent attorneys, or examiners. In-text references are references embedded in patent text, serving a very

similar role as references in scientific publications. Because of their different generation processes, front-page and in-text references embody different information and have a low overlap [4]. Furthermore, several recent studies have suggested that in-text references are a better indication of knowledge flow than front-page references [14, 3, 4].

While patent front-page references are readily retrievable from the metadata of patents, in-text references are part of the unstructured, running text. Therefore, identifying the start and end of a reference is a non-trivial task. Furthermore, patent in-text references are shorter and contain less information than front-page references (e.g. the title of the publication is typically not included), adding to the difficulty of matching in-text references to publications. For example the USPTO patent US8158424B2, "Primate pluripotent stem cells cultured in medium containing gamma-aminobutyric acid, pipecolic acid and lithium" cites a publication twice in the patent text: once as Chan et al., Nat. Biotech. 27:1033-1037 (2009) and the second time as Chan et al. Nat. Biotech 2009 Nov. 27(11):1033-7. This reference also appears as a front-page reference with more information: Chan et al., Live cell imaging distinguishes bona fide human iPS cells from partially reprogrammed cells, Nature, Biotechnology, vol. 27, pp. 1033-1037, (2009). However, most of the in-text references do not appear on the front-page and need to be extracted from the running text.

In this paper, we take up the challenges of (1) extracting references from patent texts and (2) matching the extracted references to a publication database. The second step (matching) is required because we need to uniquely identify the publications referenced in the patent for further research into the relation between science and industry.

We approach the problem of extracting in-text references as a sequence labelling task, similar to named entity recognition (NER). Sequence labelling in this regard is a supervised learning process in which each word in the text is labelled as being outside or inside a reference. We create a manually labelled training corpus and train two sequence labelling models on this corpus. We apply the models to a large corpus of 33,338 USPTO biotech patents to extract all scientific references. Once extracted, we match the extracted references to the Web of Science (WoS) database of scientific publications in a rule-based manner using regular expressions and pattern matching. We address the following research questions:

1. With what accuracy can in-text references be extracted using sequence labelling models?
2. What proportion of automatically extracted in-text references can with certainty be matched to a publication database?
3. What is the overlap between patent in-text and front-page references, and how many additional references do we discover from the full text?

We make the following contributions: (1) we deliver a solution for the challenging and unsolved problem of extracting in-text references from patents, including an annotated corpus of 22 patents;[1] (2) we show that a large number of extracted

[1] `https://github.com/tmleiden/citation-extraction-with-flair`

references can be matched to WoS publication database; (3) we show that in the biotech domain, there are a substantial number of in-text references to scientific papers that are not listed on the front page of the patent. The extraction of those in-text references will advance research into the interaction between science and innovation.

2 Related work

Matching patent front-page references Prior studies of sNPRs primarily use patent front-page references. Although front-page references are relatively easy to extract, matching them to individual publications is not a trivial task, as these references do not have a consistent format, often miss important information (e.g., author names, publication title, journal name, issue, volume, and page numbers), and are prone to errors. A number of approaches for matching front-page references to scientific publications have been proposed. Typically, the reference string is first parsed into the relevant fields: publication year, last name of the first author, journal title, volume, issue, beginning page, and article title [6, 22, 10]. Then the identified fields are matched to metadata fields of Web of Science (WoS) publications. The title is matched using string similarity metrics such as relative Levensthein distance [22, 13, 10].

Yang [22] reports a precision above 99% and recall above 95%, depending on different text similarity score thresholds. Knaus and Palzenberger [10] use the *Solr* full text search engine to retrieve WoS publications for all PATSTAT front-page references. They report a precision of 99%, and a recall of 96% and 92%, for EPO and USPTO respectively, requiring matches in at least three fields. Marx and Fuegi [13] report recall ranges from 76% to 92% and precision from 100% to 75% for different thresholds of matching scores. These methods are not directly applicable for extracting or matching patent in-text references, for two reasons. First, while front-page references are readily retrievable from the metadata in patent records, in-text references are embedded in the full text of the patent without consistent structural cues. Second, in-text references are shorter than front-page references. In particular, publication titles are rarely included, excluding the use of string similarity metrics for title overlap.

Matching patent in-text references Bryan et al. [4] developed a method for linking scientific publications and patent in-text references. Because patent in-text references are difficult to identify, they skipped this step. Instead, they started from a set of scientific publications and searched for coarse matches between meta-data of these publications and patent full texts. Specifically, they covered 3,389,853 research articles published between 1984 and 2016 in 244 prominent journals, which are cited collectively 2,779,258 times in USPTO patents granted since 1984, with 1,568,516 references of the front-page and 1,210,742 in-text.

The disadvantage of this method is that starting from scientific publications instead of patent references is computationally inefficient, considering that the WoS core collection has more than 50 million publications since 1980 and more

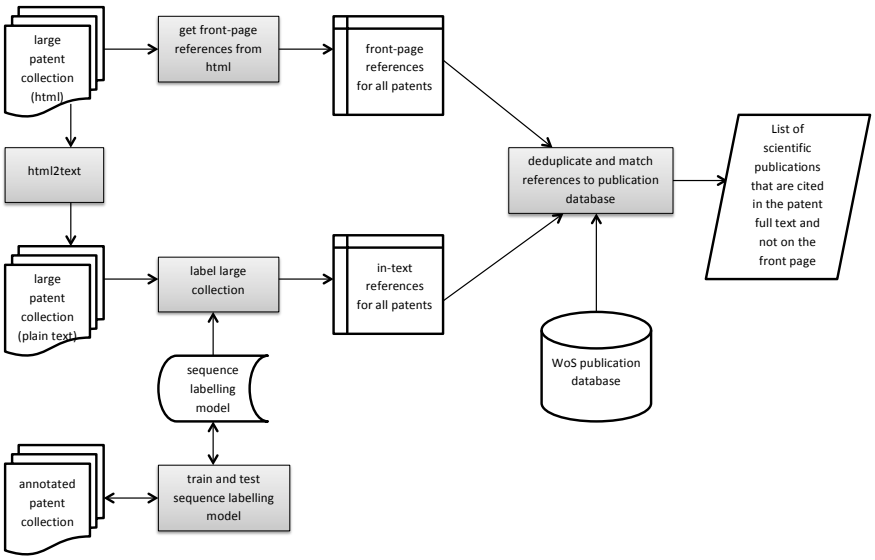

Fig. 1. Overview of the automated reference extraction and matching process. The grey blocks represent Python scripts.

than 10 thousands journals, but only a very small share of them are cited by patents: around 5% WoS publications are cited on the front-page [1, 19], and for the 244 prominent journals, 10% publications are cited on the front-page or in-text [4].

3 Methods

Our methods entail the following steps: data pre-processing and annotation, reference extraction, reference matching, and reference filtering, as illustrated in Figure 1. We will explain these steps in the subsections 3.2–3.5, after we first introduce our data in Section 3.1.

3.1 Data

We downloaded two collections of patent HTML files from Google Patents: (1) As training data for manual annotation we compiled a small collection of 22 patents with IPC class C12N, published in 2010;[2] (2) As full domain collection we obtained a larger set patents from the biotech domain, published in the years

[2] These 22 were randomly selected from the complete set of 2,365 patents with class C12N from 2010. We annotated the patents one by one until we approached 2000 references.

```
  also   encompasses nucleic-acid-like  structures  with
   O          O              O              O        O
synthetic backbones,         see           e.g.,    Mata -1997
   O          O              O              O        B    I
Toxicol.    Appl.       Pharmacol.    144:189-197;
   I          I              I              I
```

Fig. 2. Example of IOB markup for sequence labelling. The O-label indicates that a word is not part of a reference; the B-label indicates that the word is the beginning of a reference; the I-label indicates that the word is inside a reference.

2006–2010. For this second set we searched for all IPC classes associated with the biotech domain according to the OECD definition.[3] The result is a collection of 33,338 patents.

The publication database comes from Web of Science (WoS), consisting of the metadata for 22,928,875 journal articles published between 1980 and 2010 (excluding book series and non-articles, e.g., review, letter, and note). Included in this database is also a table of 19,200 journals with titles, abbreviated titles, and unique identifiers. The same unique identifiers are used in the database of publications to refer to the journal in which a paper was published.

3.2 Pre-processing and annotation

We converted the patent HTML sources to plain text using the Python package BeautifulSoup. We extracted all text in the HTML tags 'p', 'h1', 'h2', 'h3', and 'heading', excluding the text inside the tags 'style', 'script', 'head', 'title', and 'meta'. We manually annotated all in-text references in the 22 patents in the training set using the BRAT annotation tool.[4] The 22 patents contain 1,952 in-text references altogether.[5]

We converted the annotated files to IOB-format, the required format for sequence labelling methods. In IOB, each word in the text has a label B, I, or O. B means that the word is the beginning of an entity (in our case a reference), I means that the word is inside an entity, and O means that the word is not part of an entity. Figure 2 shows an example of IOB markup for a brief span of text from a patent in our hand-coded set. One difference between our problem and

[3] Query used on Google Patents: `(((A01H1/00) OR (A01H4/00) OR (A61K38/00) OR (A61K39/00) OR (A61K48/00) OR (C02F3/34) OR (C07G11/00) OR (C07G13/00) OR (C07G15/00) OR (C07K4/00) OR (C07K14/00) OR (C07K16/00) OR (C07K17/00) OR (C07K19/00) OR (C12M) OR (C12N) OR (C12P) OR (C12Q) OR (C12S) OR (G01N27/327) OR (G01N33/53) OR (G01N33/54) OR (G01N33/55) OR (G01N33/57) OR (G01N33/68) OR (G01N33/74) OR (G01N33/76) OR (G01N33/78) OR (G01N33/88) OR (G01N33/92))) country:US before:publication:20101231 after:publication:20060101 status:GRANT language:ENGLISH type:PATENT`

[4] `http://brat.nlplab.org/`

[5] The labelled data and our processing scripts are available at `https://github.com/tmleiden/citation-extraction-with-flair`

Table 1. Features used in CRF. The features 'is year', 'is name', and 'is page number' were added by us as reference-specific features.

Current token:	lowercased word (string), part-of-speech tag (string), the last 3 characters (string), the last 2 characters (string), is uppercase (boolean), starts with capital (boolean), is a number (boolean), is punctuation (boolean), is a year (boolean, pattern match), is name (boolean, list lookup), is page number (boolean, pattern match)
Context tokens (left 2, right 2):	lowercased word (string), part-of-speech tag (string), is uppercase (boolean), starts with capital (boolean), is a number (boolean), is punctuation (boolean)

named entity recognition tasks is that the references are longer than common entity types (names, places). We are interested to see to what extent the sequence labelling models can cope with these long spans.

3.3 Reference extraction

We experimented with two sequence labelling methods for reference extraction, both originally developed for named entity recognition: Conditional Random Fields (CRF) and the Flair framework.

CRF Conditional Random Fields (CRF) is a traditional sequence labelling method based on manually defined features [16]. The model finds the optimal sequence of labels (IOB) for each sentence. Each word is represented by a feature vector describing the word form, its part of speech (noun, verb, etc) and its left and right context. For part-of-speech tagging we used the maxent_treebank_pos_tagger of NLTK in Python. We used the implementation of CRF in sklearn, CRFsuite, for training the sequence labelling classifier on the hand-labelled data.[6] We extended the default feature set of CRFsuite with a few reference-specific features such as the explicit recognition of page number patterns. We included features for the 2 words before the current word and 2 words after. The feature set is shown in Table 1.

One potential limitation of CRF for reference extraction is the limited context size. This motivates the use of a method that takes a larger context into account for the labelling sequence:

Flair The Flair framework is the current state-of-the-art method for named entity recognition (NER) [2]. Flair combines a BiLSTM-CRF sequence labelling model [9] with pre-trained word embeddings. The Flair embeddings capture latent syntactic-semantic information which contextualizes words by the surrounding text. One advantage of this is that the same word will have different embedding representations depending on its contextual use. In addition, the context used in Flair embeddings is a complete paragraph (or sentence, depending

[6] `https://sklearn-crfsuite.readthedocs.io/en/latest/`

on the input format), which provides more information for the relatively long references than the limited context of CRF. The Flair framework is available online including pre-trained models.[7] The purpose of the pre-trained models is that the labelled data can be relatively small, because transfer learning is applied from the pre-trained model. It was shown in previous work that the knowledge of the pre-trained language models transfers across domains and that in small labeled datasets the use of pretrained embeddings has larger impact [18].

Flair processes the text input line by line. Unfortunately, long input lines in the training data cause the Flair framework to overload its memory.[8] To work around this, the developers advise to split paragraphs in sentences. Unfortunately, standard sentence splitting packages (NLTK, Spacy) erroneously split sentences in the middle of references because of punctuation marks in the reference text. Therefore, we decided to split sentences in the training data using the following procedure: we added a sentence split between each occurrence of a full stop and a capital letter, but only when both tokens have the O label. This way, we did not split in the middle of references. In addition, we used a minimum length of 20 tokens (to prevent non-sentences to split in short bits) and a soft maximum of 40 (to prevent memory overload).[9] In the test data however, we kept the full paragraphs instead of sentence splitting using the IOB information because this would leak ground truth labels to the test setting.

We evaluated Flair with two different embeddings models, both provided in the Flair framework distribution: the Glove embeddings for English named entity recognition [17, 20], and the English-language Flair embeddings that were trained on the 1-Billion word corpus by Chelba et al. [7, 2].[10] We adopted most parameter settings from earlier work on BiLSTM-CRF models for named entity recognition [9, 11, 2]. For the learning rate (LR), Flair uses an annealing method that halves the parameter value after 5 epochs, based on the training loss. Starting from a LR of 0.1 we assume that our model will converge (and it does) as a result of the annealing process.

Post-processing We added a post-processing step after running the sequence labelling models because sometimes multiple references are concatenated into one. This happens more often in the Flair output than in the CRF output (i.e., the beginning of references is not always marked by 'B'). Our post-processing script fixes this by splitting references on ';' if there are multiple years in the reference string with a semi-colon in between.

[7] `https://github.com/zalandoresearch/flair`

[8] This out-of-memory error is reported as known issue by the Flair developers and will be solved in a future release: `https://github.com/zalandoresearch/flair/issues/685`

[9] Our sentence splitting script is available at `https://github.com/tmleiden/citation-extraction-with-flair`

[10] Information on the embeddings models in Flair can be fount at `https://github.com/zalandoresearch/flair/blob/master/resources/docs/TUTORIAL_3_WORD_EMBEDDING.md`

Evaluation We evaluated CRF and Flair with five-fold cross validation. We split references in such a way that (a) references from the same patent are kept together in the same partition (in order to prevent over-fitting caused by similar reference contexts in the same patent); (b) the number of references is equally distributed between the partitions. Of the five partitions, three are used for training, one for validation (learning rate annealing is based on the validation set loss) and one for test, in five rotating runs.

As evaluation metrics we report Precision and Recall for the B and I labels, as well as for the complete reference. For the complete references, we do sub-string matching, where Precision is defined as the proportion of predicted references that are a substring of a true reference, and recall is defined as the proportion of true references that are found as substring of at least one predicted reference. The substring matching ensures that the presence or absence of punctuation marks at the end of reference strings do not influence the comparison.

3.4 Reference matching

A few examples of automatically extracted in-text references, illustrating their formats, are:

- Geysen et al., J. Immunol. Meth., 102:259-274 (1987)
- Altschul (1990) J. Mol. Biol. 215:403-410.
- Caohuy, and Pollard, J. Biol. Chem. 277 (28), 25217-25225 (2002);
- D. Hess, Intern Rev. Cytol., 107:367 (1987)

Matching these references to the WoS involves two steps: First, similar to prior work on front-page reference matching [6, 22, 10], we analyzed and parsed all extracted references and stored separate fields: first author, second author, year, journal title, volume/issue, and page numbers. Second, we matched these fields to WoS publications. We counted the number of matching fields to determine the strength of the match. For efficiency, we only read the WoS database once and searched for all potentially matching extracted references while reading. These are the main steps of our matching process:

1. The set R_e contains all extracted reference strings. For each $r \in R_e$:
   (a) Skip r if it does not contain one of the years 1980–2010;
   (b) Parse r to extract: last name of first author ($author_r$), last name of second author, publication year ($year_r$), journal title ($journal_r$), volume/issue, and page number;
   (c) Try to match $journal_r$ to the journal database using the abbreviated title variants in the WoS. For all references from which $journal_r$ could be extracted and matched, store the reference per journal id (the set R_j); For all references from which the journal id could not be deduced, store the reference per author (the set R_a).
2. Match references to the publication database:
   (a) Per year, read the corresponding WoS publication database. For each publication record p from this database:

i. Find references in R_j that have the same journal id as p; find the references in R_a that have the same first author as p; store them as references with possible match: the set of tuples $R_m : (r, p)$
ii. For each $(r, p) \in R_m$, count the number of additional matching fields. The maximum number of matched fields is 6: publication year, journal, pages, issue/volume, first author, second author.[11]

(b) For each $r \in R_m$, identify the best match:
- Find p with the highest number of matching fields. If at least 4 fields match then it is a strong match; if fewer fields match then it is a weak match. If there are multiple publications with the highest number of matching fields:
  - If r contains page numbers, match p that has the same page numbers;
  - If r contains page numbers but there is no p with the same page numbers, the reference does not exist in the database;
  - If r does not have page numbers, the reference is ambiguous.

(c) If there is no publication by $author_r$ in $year_r$, or by $journal_r$ in $year_r$, the reference does not exist in the database.

3.5 Reference filtering

We extracted all front-page references from the patent HTML in the metadata fields with the name attribute `citation_reference`, using the Python package BeautifulSoup. Then we searched whether each in-text reference is also listed on the front-page of the same patent, by looking up its first author and publication on the list of front-page references. This gives us information on how many additional references we can retrieve by taking the full text into account.

4 Results

We present results for reference extraction using the sample of 22 manually annotated patents (Section 4.1) and reference matching (Section 4.2) and then statistics for reference extraction and matching on the large patent collection (Section 4.3) and the combination of CRF and Flair (Section 4.4).

4.1 Reference extraction

Cross-validation results for CRF and Flair are reported in Table 2. For Flair, we found that the Flair embeddings reach higher precision and recall than the Glove embeddings, but Flair with the Glove embeddings is 30 to 40 times faster than Flair with the Flair embeddings. Flair extracts more references than CRF and a bit more than the ground truth (1,952). CRF outperforms Flair in terms of precision and recall on the B-labels and the complete references.

[11] The first author can also be a fuzzy match with an edit distance of 1. This matches author names that have a slight spelling variant in the publication database (typically a missing hyphen, e.g. SCHAEFERRIDDER for Schaefer-Ridder) or a misspelling in the reference (e.g. DEVEREUX vs. Devereaux).

Table 2. The quality of extracting patent in-text references by CRF and Flair in terms of Precision and Recall (on the label level, and on the complete references), and the number of extracted references, evaluated on 22 labeled patents using cross validation.

	B-labels		I-labels		Complete references		
Method	P	R	P	R	P	R	# of refs
CRF	**89.0%**	**82.4%**	**91.4%**	87.0%	**83.0%**	**81.3%**	1,812
Flair (Flair embeddings)	76.2%	70.2%	81.4%	**89.0%**	79.0%	75.6%	1,967
Flair (Glove embeddings)	72.2%	64.7%	78.9%	84.0%	64.7%	62.2%	2,016

Table 3. Manual validation of the matching process on a random selection of 136 matched and 275 unmatched references from the small dataset.

	#	%
Matched	136	100.0%
True positive: correctly matched	117	86.0%
Ambiguous reference text (too little information)	1	0.7%
Error in reference text (e.g. cite wrong journal, author, year)	3	2.2%
Error in reference extraction (e.g. partial or multiple references)	1	0.7%
Publication not in database (e.g. not journal)	14	10.3%
Unmatched	275	100.00%
True negative: publication not in database (e.g. not journal)	161	58.5%
Ambiguous reference text (too little information)	47	17.1%
Error in reference text (e.g. cite wrong journal, author, year)	16	5.8%
Error in reference parsing and matching	34	12.4%
Error in reference extraction (e.g. partial or multiple references)	17	6.2%

4.2 Reference matching

To evaluate the performance of our reference matching method, we manually checked 136 matched and 275 unmatched references. Table 3 shows the result of this analysis: 86.0% matched references are true positives and 58.5% unmatched references are true negatives. Our reference extraction method is only responsible for 0.7% false positives and 6.2% false negatives, and our reference matching method is responsible for 10.3% false positives and 12.4% false negatives. It is important to note that even if all in-text references are extracted perfectly with complete information, we cannot expect that all of them can be matched to WoS records. Callaert et al. [5] found that only 58% of patent front-page references are scientific. Our WoS database only includes journal articles and is a subset of what they consider as scientific. In addition, it is known of the WoS that its coverage is not fully complete. The incomplete coverage contributes to matching errors.

4.3 Application to the large collection of Biotech patents

Statistics on extracting and matching patent in-text references from the large patent collection are reported in Table 4. The Flair results were obtained using

Table 4. Statistics of patent in-text reference extracting and matching, for the large patent collection of 33,338 biotech patents. These results were obtained using the CRF and Flair (with Glove embeddings) models for the reference extraction trained on the small collection, and pattern-based matching to the WoS.

	CRF		Flair	
# of extracted in-text references from 1980–2010	519,562	100%	1,233,095	100%
# of extracted in-text references that can be parsed	484,085	93.2%	1,126,676	91.4%
" " " " with a definite match in WoS	174,899	33.7%	671,317	54.4%
" " " " with a definite match and not on the front-page	125,631		**493,583**	

Table 5. Breakdown for the extracted patent in-text references that could not be matched to WoS. These counts were generated by the matching script.

	CRF		Flair	
Total number of unmatched extracted in-text references	347,050	100%	561,778	100%
- cannot be parsed into publication fields	35,477	10.2%	106,419	18.9%
- not in the WoS publication database	35,477	16.0%	267,208	47.6%
- ambiguous reference	254,218	73.8%	188,989	33.6%

the Glove embeddings because the Flair embeddings were 30 to 40 times slower in generating output (see Section 4.1). Table 5 presents the breakdown for the unmatched in-text references. The number of patents in the collection is 33,338; altogether they have 1,174,661 front-page references. CRF extracts 125,631 in-text references that are not on the front-page; Flair extracts much more: 493,583.

Table 4 and 5 show large differences between CRF and Flair. A much larger number of references extracted by Flair can be matched to WoS than that of CRF. For CRF, the majority of references without a definite match are ambiguous, meaning that they have multiple possible matches in WoS. One example is: "Sunamoto et al. (Bull. Chem. Soc. Jpn., 1980, 53,". There are five records in WoS with Sunamoto as the first author, multiple other authors, the same journal, year, and volume, three of which even appear in the same issue. Without additional information about other authors, issue, and page numbers, it is impossible to know which publication is actually being referenced. Further analysis of these cases indicated that the ambiguity occurs because the disambiguating information is not part of the reference text extracted by CRF: For the majority (72%) references extracted by CRF, only 2 of the 5 most important fields (first author, year, journal id, issue, page numbers) can be extracted through parsing the references, while for the majority (72%) references extracted by Flair, at least 4 of those fields can be extracted.

4.4 Combining the output of Flair and CRF

To assess the total overlap between in-text and front-page references and added information value of in-text references, we combine Flair and CRF outputs. Flair and CRF collectively extracted 686,956 in-text references from the large patent

collection that could matched to the WoS publications, and 603,457 (88%) of those are not listed on the patent front-page.

The collection of 33,338 Biotech patents contains 1,174,661 non-patent front-page references in total. The additionally retrieved 603,457 in-text references constitute a 51% increase in identified patent-publication-links, which is a conservative estimation considering that only around 58% front-page references are actually scientific [5].

5 Conclusion

This paper tackles the challenge of extracting and matching patent in-text references to scientific publications. We approach the reference extraction problem as a sequence labelling task using CRF and Flair. We solve the reference matching problem in a rule-based manner, using regular expressions for extracting publication fields and then matching them to the Web of Science database. Specifically, We trained the models and developed the patterns on a small, manually labelled sample of 22 patents with 1,952 references. Then we applied the models to a large collection of 33,338 biotech patents.

(RQ1) We trained two supervised models on the manually annotated sample. CRF achieved the best result in cross validation: for individual B and I labels, precision scores are 89% and 91% respectively, and recall scores 84% and 87% respectively. For complete references, precision is 83% and recall 81%. The state-of-the-art sequence labelling method Flair did not beat CRF on most of the evaluation metrics (only Recall for the I-labels). This is probably due to a mismatch between train and test settings for sentence splitting in Flair, necessitated by known memory issues of the framework. We are currently investigating whether we can improve our Flair model by fine-tuning the language model on domain data (i.e., biotech patents).

(RQ2) Our method is able to match a large number of the extracted references to WoS publications. CRF extracted 519,562 in-text references from the years 1980–2010 from the large patent collection, 33.7% (172,899) of which had a definite match in the WoS publication database. Flair extracted much more references (1,233,095), and 54.4% of them (671,317) had a definite match. Thus, although Flair is less exact in extracting references than CRF, it extracts more references, and more of those can be matched to WoS publication records, because the extracted strings are more complete (reflected by a higher recall for the I-labels).

(RQ3) Flair and CRF collectively matched 686,956 in-text references from the large patent collection to WoS, and 603,457 (88%) of those are not listed on the patent front-page. These additionally retrieved references constitute a substantial increase (51%) compared to the set of front-page references. These findings highlight the added value of patent in-text references for studying the interaction between science and innovation.

6 Acknowledgements

We thank Yuanmin Xu for making the manual annotations.

References

1. Ahmadpoor, M., Jones, B.F.: The dual frontier: Patented inventions and prior scientific advance. Science **357**(6351), 583–587 (2017)
2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1638–1649. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1139
3. Bryan, K.A., Ozcan, Y.: The impact of open access mandates on invention. Mimeo, Toronto (2016)
4. Bryan, K.A., Ozcan, Y., Sampat, B.N.: In-text patent citations: A users guide. Tech. rep., National Bureau of Economic Research (2019)
5. Callaert, J., Grouwels, J., Van Looy, B.: Delineating the scientific footprint in technology: Identifying scientific publications within non-patent references. Scientometrics **91**(2), 383–398 (2011)
6. Callaert, J., Vervenne, J.B., Van Looy, B., Magerman, T., Song, X., Jeuris, W.: Patterns of science-technology linkage. Tech. rep., European Commission (2014)
7. Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., Robinson, T.: One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005 (2013)
8. Fleming, L., Sorenson, O.: Science as a map in technological search. Strategic Management Journal **25**(8-9), 909–928 (2004)
9. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
10. Knaus, J., Palzenberger, M.: Parma. a full text search based method for matching non-patent literature citations with scientific reference databases. a pilot study. Tech. rep., Max Planck Digital Library (2018)
11. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
12. Li, D., Azoulay, P., Sampat, B.N.: The applied value of public investments in biomedical research. Science **356**(6333), 78–81 (2017)
13. Marx, M., Fuegi, A.: Reliance on science in patenting. SSRN: https://ssrn.com/abstract=3331686 (2019)
14. Nagaoka, S., Yamauchi, I.: The use of science for inventions and its identification: Patent level evidence matched with survey. Research Institute of Economy, Trade and Industry (RIETI) (2015)
15. Narin, F., Hamilton, K.S., Olivastro, D.: The increasing linkage between us technology and public science. Research policy **26**(3), 317–330 (1997)
16. Okazaki, N.: Crfsuite: a fast implementation of conditional random fields (crfs) (2007)
17. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
18. Peters, M.E., Ammar, W., Bhagavatula, C., Power, R.: Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108 (2017)

19. Poege, F., Harhoff, D., Gaessler, F., Baruffaldi, S.: Science quality and the value of inventions. arXiv preprint arXiv:1903.05020 (2019)
20. Reimers, N., Gurevych, I.: Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. arXiv preprint arXiv:1707.09861 (2017)
21. Veugelers, R., Wang, J.: Scientific novelty and technological impact. Research Policy **48**(6), 1362–1372 (2019)
22. Yang, S.: Linking Science and Technology: Reference Matching for Co-citation Network Analysis. Master's thesis, Leiden University (2016)