Exploring topic structure: Coherence, diversity and relatedness
He, J.

Citation for published version (APA):
He, J. (2011). Exploring topic structure: Coherence, diversity and relatedness

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 9

Automatic Link Generation for Radiology Reports

In the previous chapter, we empirically analyzed several factors that have an impact on automatic link generation with Wikipedia. In this chapter, we move from linking Wikipedia topics towards linking free texts to Wikipedia. The main research question we address in this chapter is as follows.

**RQ7** Can state-of-the-art ALG systems that are, in principle, domain independent, be effectively applied to linking texts from a specific domain to Wikipedia? If not, can we improve the effectiveness of automatic link generation by considering domain specific properties of the data?

More specifically, we study a case where we use automatic link generation technology to annotate radiology reports with Wikipedia topics. Within this context, more specific research questions related to this main question are raised. See Section 9.1 below.

9.1 Introduction

We start by providing some background. Two trends are influencing the role of radiology in the care process. First, the services delivered by radiologists are becoming a commodity, that is, they can be delivered by any radiology party “without qualitative differentiation across [the] market”.\(^1\) This trend is caused by various technological advances and societal trends, such as teleradiology, picture archiving and communication systems, computer-aided diagnosis software, communication standards, and an increasing demand for cost effectiveness [21, 26, 169, 194]. A lively debate has ensued whether the commoditization of radiology is a desirable trend, and how it can be directed to safeguard the quality of care and the role of radiology in the care process [21, 22, 23, 68, 123, 136]. Second, the practice of radiology is influenced by the shift in medicine from a provider-centric model of care to a patient-centric model [117, 118].

This shift calls for improved and novel ways for radiologists to communicate with patients [169], which is especially challenging for radiology, as its practitioners typically have no direct contact with patients [193] (except for some subdisciplines, such as interventional radiology).

Both trends call for means to increase the value of radiology in the care chain, especially the value perceived by the patient, and preferably without increasing the radiologist’s workload. The most important contribution of radiology to the care process are interpretations of radiology images that are communicated through narrative reports to, primarily, colleague clinicians [184, 195]. Various ways have been proposed to increase the economic value of reports, such as restructuring their contents [184], adding citations to the medical literature and embedding key images. These enhancements aim at increasing the value from the referring clinicians’ point of view, but they do not necessarily serve the patients’ interests [193].

In this chapter, we introduce a way to enhance radiology reports by adding links to Wikipedia. This scenario gives rise to the following tasks: given a radiology report, (i) mark the relevant medical phrases, and (ii) for each phrase marked, generate a link to a Wikipedia page that provides background information about the phrase. As defined in the context of Automatic Link Generation (ALG) (see Chapter 8), we refer to the first task as anchor detection, where the marked medical phrases are anchor texts, and refer to the second task as target finding. It is envisioned that these explanatory links, rendered as hyperlinks in the report, help the patient to understand the clinical vocabulary and the implications the report has for his or her medical situation. This will help to empower the patient in the care process and to reduce anxiety. Since the proposed system generates hyperlinks without human intervention, the annotation process does not put additional pressure on the radiologist’s workload.

In principle, the techniques described in this chapter can be applied to any other medical knowledge source such as MedlinePlus, produced and maintained by the National Library of Medicine. It can also be used to support other stakeholders such as referring physicians who may find it more useful to consult expert-level sources, for example Amirsys’ STATdx encyclopedia. Wikipedia is chosen mainly for the following reasons.

**Quantity**  Wikipedia densely covers the medical domain. It contains medical lemmas from multiple medical thesauri and ontologies, for example International Statistical Classification of Diseases and Related Health Problems (ICD-9, ICD-10), Gray’s Anatomy, etc.

**Quality**  Although Wikipedia is written collaboratively by largely anonymous Internet volunteers, the quality of articles is guaranteed by the Wikipedia content criterion “verifiability,” that is, readers should be able to verify the material in a Wikipedia

---

2\text{http://www.nlm.nih.gov/medlineplus}

3\text{http://www.amirsys.com}
page against a reliable source. In addition, errors in the content are often spotted quickly and corrected by collaborative editors [247].

**Accessibility** Wikipedia is a free online resource. All users can access its content without registering or creating an account. Moreover, the content of Wikipedia is usually written at a level understandable for patients, i.e., non-experts.

**Maintenance** The discussion tabs of a medical Wikipedia page generally contain a wealth of information that also documents changes to earlier version of the page.

Most of the studies in automatic link generation focus on solving a general problem, (e.g., developing an automatic link generation approach using Wikipedia as training material and applying it to any topic domain [175, 178]), or applying the link generation techniques in general domains that cover diverse sets of topics, (e.g., news, blogs, web, etc. [53, 126, 173]). Here, we focus on applying automatic link generation techniques to data from the radiology domain.\(^4\)

Two features set radiology data apart from data from a general domain. On the one hand, medical phrases often have a regular syntactic structure. For example, they are often noun phrases with one or more modifiers (e.g., adjectives). Such regularity provides useful features for recognizing these medical phrases in the reports. On the other hand, in many cases, the presence of multiple modifiers as well as conjunctions within a single medical phrase results in a complex semantic structure. In other words, a complex topic structure in the context of this thesis. For example, the phrase “acute cerebral and cerebellar infarction” contains two topics “cerebellar infarction” and “acute cerebral infarction,” where “cerebellar” and “cerebral” are synonyms. When linking this phrase to Wikipedia, one needs to identify the main topic it represents prior to searching for a target page in Wikipedia.

With the above mentioned properties in mind, we aim to develop an approach that takes into account the domain properties of the radiology reports, so that the effectiveness of link generation on radiology reports can be improved over those state-of-the-art systems that are developed for data from a general domain. Specifically, we seek answers to the following research questions:

**RQ7a** How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

**RQ7b** How does our proposed approach compare to state-of-the-art approaches aimed at solving the automatic link generation problem in the general domain?

In addition, we notice that some medical phrases are more frequently seen than others. For example, “brain,” as a relatively common topic in neuro-radiology, appears in almost all neuro-radiology reports, while “xanthogranulomas” only occurs in reports

\(^4\)A radiology report is a semistructured document that in general consists of the following components: clinical information, e.g., symptoms, procedure of the radiology scan, findings from the scan, and impression, i.e., the opinion of the radiologist.
that discuss this specific medical condition. Here, we investigate how the frequencies of medical phrases are distributed over radiology reports and whether automatic link generation systems perform differently when dealing with medical phrases with different frequencies. Further, if there is a difference, how does the difference affect the overall performance of an automatic link generation system? Consequently, we formulate our third research question as

**RQ7c** What is the impact of anchor text frequency on the performance of automatic link generation systems?

We seek the answers to our research questions using empirical methods. A test collection is manually created for this investigation (see Section 9.3).

The contribution of the chapter can be summarized as follows. First, we propose an automatic link generation approach that aims at enhancing narrative radiology reports by automatically adding links from medical concepts in the reports to Wikipedia. This approach is shown to improve over two state-of-the-art link generation systems that have previously been developed to solve the automatic link generation problem in a general domain. Second, we conduct an in-depth analysis of the performance of both state-of-the-art systems and our proposed approach. The conclusions of our analysis provide useful hints for future work on this research topic.

The remainder of the chapter is organized as follows. Section 9.2 discusses related work in information extraction and mapping in the biomedical domain and applications in the radiology domain. In Section 9.3, we describe two state-of-the-art automatic link generation systems, which serve as baseline systems. In Section 9.4, we introduce our approach to automatically generate links for narrative radiology reports. Then in Section 9.5 we specify our experimental setup for evaluating our proposed approach. Section 9.6 compares the experimental results of our approach to that of the state-of-the-art systems, followed by a discussion on the factors that cause the difference in system performance in Section 9.7. In Section 9.8 we analyze the impact of anchor text frequency on the performance of automatic link generation systems. Section 9.9 concludes the chapter with answers to the research questions and a discussion of future directions for our work.

### 9.2 Information extraction and mapping for biomedical data

Natural language processing techniques have been widely applied in the biomedical domain to disclose information from clinical free-text documents. We highlight two tasks from medical natural language processing that are related to our work, namely, biomedical named entity recognition (NER) and concept mapping.

The NER task addresses identification of biomedical terminology, for example gene or protein names, from free text such as biomedical literature. This task is
very similar to the anchor text identification task we discuss in this chapter, which aims at identifying anatomy and diagnosis terms from radiology reports. The major biomedical NER methods fall into three categories [134]: dictionary-based approaches [5, 135, 220, 243, 244, 264], rule-based approaches [9, 70, 73, 74, 109, 240] and machine learning methods [45, 129, 148, 170, 224, 238, 273].

Compared to other types of approach, machine learning approaches have the advantage of being robust and flexible, as they generally generalize well beyond given vocabularies and easily adapt to new language styles. The machine learning techniques most commonly used in this area include the ones that are well-known for solving sequential labeling problems, such as Hidden Markov Models (HMM) [45, 273], Support Vector Machines (SVM) [129, 148, 238] and Conditional Random Fields (CRF) [170, 224]. Various types of feature have been explored, particularly syntactic features such as part-of-speech (POS) tags and orthographical features such as the combination of digits and letters. This is due to the fact that biomedical terminology, such as gene and protein names, often displays syntactic regularities as well as uncommon word spellings. In this chapter, we apply a CRF-based sequential labeling approach to our anchor text identification problem, as we have noticed that similar to gene and protein names, the annotated anchor texts in radiology reports display strong syntactic regularities.

The concept mapping task focuses on mapping names to concepts in a reference biomedical ontology, such as the Unified Medical Language System (UMLS)\footnote{http://www.nlm.nih.gov/research/umls/} and Medical Subject Headings (MeSH).\footnote{http://www.ncbi.nlm.nih.gov/mesh} These ontologies attach one or more descriptions to each concept and interrelate concepts through a number of relation types. For a given biomedical name, the step of finding the most appropriate concept in the reference ontology resembles the target finding task of automatic link generation. Representative systems include MetaMap [10] and Peregrine [226]. In the development of these systems, much effort has been devoted to resolving term variations and term ambiguity.

Some research programs have taken a more implicit viewpoint on concept mapping, in the sense that they do not map an explicit biomedical name to a concept from a reference ontology, but the entire body of text that contains the name. This type of mapping usually uses information retrieval techniques to rank the concepts from the reference data source in descending order of their relevance to the input text. For example, the EAGL system proposed by Ruch [206] assigns MeSH concepts to an input text using a retrieval system based on vector space models, and Trieschnigg et al. [242] use a retrieval system based on language models.

In the radiology domain, a number of information extraction systems have been developed that focus on narrative radiology reports. The Special Purpose Radiology Understanding System (SPRUS) [91] is one of the earlier systems of its kind that extracts and encodes findings and interpretations from chest radiology reports. The authors also experiment with syntactic extensions of SPRUS, reporting a 81% recognition rate in a small scale experiments (10 reports) [92]. The Medical Language Extraction and
Encoding System (MedLee) [71] is a rule-based system designed to extract clinical information from clinical radiology reports and encode them in terms of a controlled vocabulary. It reports 70% recall and 87% precision scores on identifying four diseases from a set of 230 radiology reports. Recently, Soysal et al. [233] have proposed the Turkish Radiology Information Extraction System (TRIES) that extracts and converts clinical information from Turkish radiology reports based on manually crafted rules and a domain specific ontology. The authors report 93% recall and 98% precision scores on a corpus of abdominal radiology reports. The high performance, as stated by the authors, is mainly due to the effectiveness of the hand-crafted rules and the rich morphological structure of the Turkish language [233].

While all systems discussed above map terms to certain ontologies or thesauri, we aim to map the identified anchor texts to Wikipedia pages. Further, while our proposed system uses a machine learning based approach that does not require any external knowledge sources such as an ontology or hand crafted rules, it is flexible enough to be extended with this type of domain specific expert knowledge.

9.3 Two state-of-the-art automatic link generation systems

In this section, we discuss two state-of-the-art automatic link generation systems, namely Wikify! [175] and Wikipedia Miner [178]. We continue to use the notation as specified in Section 8.2.1 on page 137.

The procedure of automatically generating links from free text to Wikipedia can be divided into the following three components: (1) anchor detection (AD); (2) target candidate identification (TCI); and (3) target detection (TD). Note that TCI and TD together can be seen as the target finding task. Here we decompose this task into two components because our proposed approach introduced in Section 9.4 has a major difference in the TCI component compared to the state-of-the-art systems introduced in this section. We define the following three functions corresponding to the three components discussed above: $AD(\cdot)$ detects a set of anchor texts $A^t$ from the set of ngrams $NG^t$ extracted from $t$; $TCI(\cdot)$ collects candidate target pages $C^a$ from $W$ with respect to an anchor text $a$, and $TD(\cdot)$ finds the target page $d^a$ from $C^a$ for $a$, i.e., identifies links $L^t = \{l_i(a, d^a)\}_{i=1}^{L^t}$.

9.3.1 Wikify!

The procedure by which the Wikify! system generates links from a source text $t$ to a Wikipedia page can be summarized in the pseudo code illustrated in Algorithm 2. Below, we briefly describe the approaches the Wikify! system uses to implement the three functions $AD(\cdot)$, $TCI(\cdot)$ and $TD(\cdot)$. 
Algorithm 2 Workflow of Wikify!

Input: \( NG^t \)
Output: \( L^t \)
\( A^t = \emptyset, L^t = \emptyset. \)
\( A^t = AD(NG^t) \)
for \( a \) in \( A^t \) do
    \( C^a = TCI(a, W) \)
    \( d^* = TD(C^a, a) \)
    \( L^t \leftarrow L^t \cup \{(a, d^*)\} \)
end for
return \( L^t \)

Anchor detection  For detecting anchor texts from \( NG^t \), the Wikify! system ranks each ngram \( ng \in NG^t \) according to a score and uses the top \( \tau \) ranked \( ng \)'s as anchor texts for \( t \). Mihalcea and Csomai [175] have experimented with several scores, including TF.IDF, \( \chi^2 \) and a keyphraseness score which turns out to be the most effective score among the three. The keyphraseness score is defined as follows.

\[
keyphraseness = \frac{|A_{ng}|}{|D_{ng}|} \quad (9.1)
\]

where \( |A_{ng}| \) is the number of Wikipedia pages where \( ng \) occurs as an anchor text, and \( |D_{ng}| \) is the number of Wikipedia pages that mention the ngram \( ng \).

Target candidate identification  The Wikify! system collects \( C^a \) for a given \( a \) via existing Wikipedia links. That is, in Wikipedia, when an ngram is used as an anchor text in a source text, there exists a target Wikipedia page it links to, and this target page is selected as a candidate page. If the ngram has multiple interpretations, then different occurrences of this ngram may be linked to different target pages in Wikipedia, depending on the context of the occurrences.

Target detection  To identify the target page \( d^* \) from \( C^a \) for a given \( a \), Mihalcea and Csomai [175] have experimented with two approaches. The first one is a knowledge based approach, which selects the candidate target page that maximizes a score calculated using the Lesk [151] algorithm as the target page. The Lesk algorithm is used to calculate the word overlap between the candidate target page and the context where \( a \) occurs. The second approach uses a machine learning based approach. For each \( a \), a classifier is trained to classify whether a candidate target page should be linked.

9.3.2 Wikipedia miner

The Wikipedia miner system implements the approaches proposed by Milne and Witten [178]. We summarize the workflow of the Wikipedia miner system using the
pseudo code in Algorithm 3.

**Algorithm 3** Workflow of Wikipedia miner

```
Input: NG
Output: L
A' = Φ, L' = Φ, Atmp = Φ.
for ng in NG do
    C_{ng} = TCI(ng,W)
    d* = TD(C_{ng}, ng)
    Atmp ← Atmp ∪ (ng, d*)
end for
A' = AD(Atmp)
for a in A' do
    L' ← L' ∪ {l(a, d*)}
end for
return L'
```

**Target candidate identification**  The Wikipedia miner system differs from the Wikify! system in that target candidate identification and target detection is performed over ngrams in stead of identified anchors. To collect target candidates, Wikipedia miner uses existing Wikipedia links. To improve efficiency, a threshold is used to filter out candidate pages that have very low chance of being linked to a given ngram ng based on the observations made from links among Wikipedia pages.

**Target detection**  Wikipedia miner trains a classifier for target detection. One important feature is the relatedness of a candidate c to the context terms of an ngram ng. Specifically, a context term of an ngram ng is defined as an ngram that co-occurs with ng in t and is always linked to the same target page for all its occurrences within Wikipedia. A relatedness score is calculated to measure the semantic similarity of c and the target page of a context term by comparing their incoming and outgoing links. For more details of the features as well as the combination of features employed in the Wikipedia miner system, we refer to [178].

**Anchor (link) detection**  Wikipedia miner does not have an explicit “anchor detection” phase, instead, anchor detection is achieved by detecting \((a, d^*_a)\) pairs from all \((ng, d^*_ng)\) pairs. Hence, the result of anchor detection is a set of links, since anchor texts are found together with their targets and each pair is classified as either “link” or “not a link.” A classifier is trained over instances consisting of ngram-target pairs. Various features are used to train the classifier, including the keyphraseness score proposed in [175] and features reflecting the relatedness between source text and target page.
9.4 Method

We now proceed to introduce our approach. In general, the workflow of our own proposed system is the same as that of the Wikify! system, as illustrated in Algorithm 2. That is, it follows the following steps: anchor detection, target candidate identification and target detection.

9.4.1 Motivation

As discussed in Section 9.1, we have identified two properties of the anchor texts in radiology reports: the regularity of their syntactic structure and the complexity of their semantic structure.

In order to exploit the regularity of syntactic structure of medical phrases in the radiology reports, we treat the anchor detection problem as a sequential labeling problem. Sequential labeling is an effective approach in terminology recognition in various applications in the biomedical domain [45, 129, 148, 170, 224, 238, 273]. In addition, different from the two state-of-the-art systems, we learn the pattern of anchor texts from radiology data instead of Wikipedia. Intuitively, the anchor texts in Wikipedia are from a general domain and have a different syntactic structure from the medical anchor texts in the radiology data, therefore they may not provide effective training material for sequential labeling.

To cope with the complex semantic structure of the medical anchor texts, we propose a sub-anchor-based approach to retrieve candidate targets and to formulate features for target detection. By retrieving target candidates with respect to sub-anchors of an anchor text, we collect candidate pages that are potentially relevant to different topics contained in the anchor text. Then at the target detection phase, we aggregate features extracted at sub-anchor level to anchor-level. The feature of a single sub-anchor text is weighted by the importance of that sub-anchor, which is measured by its similarity to the original anchor text.

In the rest of this section, we first describe our approaches to the three components mentioned above. Then we summarize our approaches by giving an overview of our link generation system LiRa, which integrates these components.

9.4.2 Anchor detection

We define the sequential labeling task for anchor detection as follows. Given a text document, identify anchor texts by annotating each of the words in the text with one of the following labels: begin of anchor (BOA), in anchor (IA), end of anchor (EOA), outside anchor (OA), and single word anchor (SWA). SWA defines a single word anchor; BOA-(IA)-EOA defines an anchor with multiple words. Within this framework, we will use a conditional random fields (CRF) model [144], which has shown state-of-the-art performance in solving sequential labeling problems [170, 224].
Let \( WS = w_1, \ldots, w_n \) be an observed word sequence of length \( n \), and \( SS = s_1, \ldots, s_n \) a sequence of states where \( s_i \) corresponds to the label assigned to the word \( w_i \). Following Settles [224], we use linear-chain CRFs, which define the conditional probability of the state sequence given the observed word sequence as

\[
p(SS|WS) = \frac{1}{Z(WS)} \exp \sum_{i} \sum_{k} \lambda_k f_k(s_{i-1}, s_i, w_i, i),
\]

(9.2)

where \( Z(WS) \) is a normalization factor over all state sequences, \( f_k(\cdot) \) is a feature function and \( \lambda_k \) is a learnt weight for feature \( f_k(\cdot) \). The feature function describes a feature corresponding to the position \( i \) of the input sequence, states at position \( i \) and \( i-1 \), and word at position \( i \).

The goal of the learning procedure is to find the feature weights \( \lambda \) that maximize the log-likelihood of the training data:

\[
LL = \sum_i \log p(s_i|w_i) - \sum_k \frac{\lambda_k^2}{2\sigma^2}.
\]

(9.3)

The second term in Eq. 9.3 is a spherical Gaussian weight prior [38] used to penalize the log-likelihood term to avoid over-fitting.

We use three simple features: the word itself, its part-of-speech (POS) tag and its syntactic chunk tag. We have also conducted preliminary experiments with several variations of these features, including orthographical features of the word, e.g., whether it contains digits, capitalization, as well as bigram and trigram features. However, the performance of these variations in anchor detection in terms of both precision and recall (see 9.5.3) are negligibly close to that of the three basic features. Therefore we focus on the three basic features.

### 9.4.3 Target candidate identification

#### Anchor decomposition

Given an identified anchor text \( a \) of length \( l \), we decompose it into a set of all sub-sequences \( Sa = \{s_i\}_{i=1}^m \), while keeping the original order of the words within the identified anchor text. For example, for the anchor text “white matter disease”, we have a set of sub-sequences \{“white”, “matter”, “disease”, “white matter”, “matter disease”, “white disease”, “white matter disease”\}. We call those sub-sequences sub-anchors.

In addition, one feature of Wikipedia is that there exist redirect pages, which provide synonyms or morphological variations for a concept. For example, the concept “acoustic schwannoma” is redirected to “vestibular schwannoma.” While decomposing an identified anchor text, we add those redirects to the set of sub-anchors, in order to reduce term mismatching and thus increase the recall of the annotated targets.
Candidate target retrieval

For each sub-anchor $s$, we retrieve a set of candidate target pages $C_s = \{c_i\}_{i=1}^n$, ranked in descending order of their target probability. Let $L_{s,c} = \{l(a,d^*)|a = s, d = c, d \in W\}$ denote all pairs of links found between $s$ and $c$ in Wikipedia links, that is, links between target page $s$ and all occurrences of $s$ as anchor texts. The target probability is calculated as

$$p(c_i|s) = \frac{|L_{s,c_i}|}{\sum_{j=1}^n |L_{s,c_j}|}. \quad (9.4)$$

We collect the top-$K$ Wikipedia pages in terms of their target probability scores for each sub-anchor and use the union of all the collected pages from each sub-anchor as the candidate target pages for the anchor. When examining the occurrence of sub-anchor $s$ in existing Wikipedia links, we consider partial matches of phrases. That is, if all terms in $s$ appear ordered within a Wikipedia anchor text, it is considered to be an occurrence. In addition, if the title of a Wikipedia page matches $s$, we also include this page as a candidate target page.

9.4.4 Target detection

We use a machine learning based approach to identify the target page $d^*$ for a given anchor text $a$. Specifically, we train a classifier over the anchor-target–candidate pairs $(a,c)$, which are labeled as “link” or “non-link”. We extract the following features to train the classifier: (i) title matching, (ii) target probability, and (iii) language model log-likelihood ratio. The first two features are calculated at the sub-anchor level, and the third feature is calculated for a candidate target page. Below, we explain each of the features.

Title matching

We consider the title matching scores for each of the sub-anchors. For a sub-anchor $s$ of anchor $a$, and a candidate target page $c$, the title matching score is defined as follows:

$$tm(s,c) = f_{tm}(s,c) \frac{\text{len}(s)}{\text{len}(a)}, \quad (9.5)$$

where

$$f_{tm}(s,c) = \begin{cases} 1 & \text{if } s \text{ equals title of } c \\ 0 & \text{otherwise.} \end{cases}$$

and $\text{len}()$ is number of words in a word sequence.

The title matching score reflects the degree of matching between the anchor text and the title of $c$. The longer the sub-anchor, the more similar the sub-anchor is to the original anchor text, and therefore we have a higher degree of matching between the anchor text and the title of $c$. 
Target probability

As defined in Eq. 9.4, the target probability is the probability that a Wikipedia page will be selected as target page, given the anchor text. We calculate $p(c|s)$ for each sub-anchor $s$, which is in fact precomputed during the candidate retrieval procedure.

Since the target probability is calculated at the sub-anchor level, we need to aggregate those scores for the original anchor texts. Note that in the case of title matching, no explicit aggregation is needed, since for a given candidate target page, it can only match one of its sub-anchors. In the case of candidate target probability, we aggregate the features extracted from the sub-anchors into three features. For an anchor $a$ and its sub-anchors $S_a$ of a candidate target page $c$ we define:

\[
\begin{align*}
\text{max}_{tp}(c) &= \max_{s \in S} p(c|s); \\
\text{min}_{tp}(c) &= \min_{s \in S} p(c|s); \\
\text{wsum}_{tp}(c) &= \sum_{s \in S_a} \frac{\text{len}(s)}{\text{len}(a)} p(c|s).
\end{align*}
\]

Language-model log-likelihood ratio (LLR)

The language-model log-likelihood ratio feature indicates to which extent a candidate target page is about radiology.

Language models are statistical models that capture the statistical regularities in generating a language [190]. Here we consider two language models. The first, $\theta_R$, models the language used in the radiology reports, which we refer to as the radiology model, and the second, $\theta_W$, models the language used in Wikipedia pages on topics in a general domain, which we refer to as Wikipedia model.

Each model defines a probability mechanism, which can be explained as follows. Assuming the two models sample terms from the radiology collection and the Wikipedia collection that follow a multinomial distribution, using a maximum likelihood estimation, the probability that a certain term $t$ is selected given a collection can be estimated as the relative frequency of the term in the collection. Now, given a piece of text with $n$ terms, $T = \{t_i\}_{i=1}^n$, the two models repeatedly sample $n$ times, assuming independence between successive events. The probability that $T$ is generated by the radiology language model can be defined as

\[
p(t_1, t_2, \ldots, t_n|\theta_R) = \prod_{i=1}^n p(t_i|\theta_R), \tag{9.6}
\]

while the probability that $T$ is generated by the Wikipedia language model is

\[
p(t_1, t_2, \ldots, t_n|\theta_W) = \prod_{i=1}^n p(t_i|\theta_W). \tag{9.7}
\]
Given the above language models, we use the log-likelihood ratio (LLR) [166], a widely used model-comparison metric, to decide which model is more likely to have generated $T$:

$$LLR(T) = \log \left( \frac{p(T|\theta_R)}{p(T|\theta_W)} \right) = \frac{\sum_{i=1}^{n} \log p(t_i|\theta_R)}{\sum_{i=1}^{n} \log p(t_i|\theta_W)}.$$  \hspace{1cm} (9.8)

To avoid zero probabilities, which come up if terms in $T$ do not occur in the radiology reports or in Wikipedia, we use Laplacian smoothing [152]. That is, we assume that each word has been seen at least once.

The LLR score indicates which of the two models $\theta_R$ and $\theta_W$ is most likely to have generated $T$. A score larger than 0 indicates $T$ is more likely to be generated by the radiology language model, hence more likely to be relevant to the anchor text identified from a radiology report.

In summary, we list the final features we use to train a classifier for identifying a target page from a set of candidate targets:

1. Title matching between $a, c$;
2. Maximum target probability $\max_{t} p$;
3. Minimum target probability $\min_{t} p$;
4. Weighted sum of target probability $w\sum_{t} p$;
5. Language model log-likelihood ratio of $c$.

### 9.4.5 LiRa: a system overview

In Figure 9.1, we show an overview of the architecture of the proposed system LiRa for automatically generating links from radiology reports to Wikipedia. When LiRa receives a radiology report, it first parses the report and extracts the features needed for sequential labeling. After sequential labeling, the identified anchor texts are passed to the next stage for target detection. For each anchor text, LiRa retrieves a set of candidate target pages, extracts features and submits to the trained classifier. The output of the classifier is aggregated and generates the final annotated reports.
9.5 Experiments

9.5.1 Research questions and experimental setup

Recall the research questions raised in Section 9.1:

RQ7a How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

RQ7b How does our proposed approach compare to state-of-art approaches that are aimed at solving the automatic link generation problem in the general domain?

RQ7c What is the impact of anchor text frequency on the performance of automatic link generation systems?

To answer RQ7a, we offer our proposed automatic link generation approach as described in Section 9.4. We evaluate our approaches on the test collection that is developed on the purpose of evaluating automatic link generation for radiology reports. We describe the details of the collection in 9.5.2. We evaluate the systems on three aspects: (i) anchor text detection; (ii) target finding; and (iii) the overall performance of the system in generating the links.

For RQ7b, we run the two state-of-the-art link generation systems, namely Wikify! and Wikipedia miner, on the same test collection and compare their results against
9.5. Experiments

the results of our proposed approach. We discuss the results and comparisons in Section 9.6. Note that in order to compare the performance of systems in target finding, we need to run the target finding components of each system on a same set of anchor texts. We include two sets of anchor texts for evaluation. First, the annotated anchor texts found in the ground truth can be used for this purpose. However, since we run the Wikipedia miner system as a black box (see Section 9.5.5), we do not have access to the intermediate result of target finding. Therefore this set can only be used to compare our system against the Wikify! system. The second anchor text set we consider is the anchor texts identified by the Wikify! system or by Wikipedia miner. That is, we run LiRa on the anchor texts identified by Wikify! (Wikipedia miner), and compare the target finding performance of LiRa against that of Wikify! (Wikipedia miner) on the same set of anchor texts.

On top of that, we provide two rounds of analysis. The first analysis, described in Section 9.7, further investigates the difference between the state-of-the-art systems and our proposed system in terms of their effectiveness of identifying anchors and linking them to the correct target pages. Particularly, we focus on the factors that make a system effective or non-effective.

The second analysis aims at answering RQ7c, where we compare the performance of the systems in anchor detection and target detection with respect to anchors with different frequencies. See Section 9.8.

9.5.2 Test collection

Our test collection is based on 860 deidentified neuroradiology reports, obtained from a US-based radiology institute. For the sake of the annotation process, the corpus was divided in three subsets; each subset was assigned to an annotator. Each annotator manually selected the anatomy and diagnosis phrases (i.e., the anchor texts in our experiments) in all reports assigned to him. The selections were stored as character ranges. Selections were allowed to overlap. For example, in the string “vestibular schwannomas,” both “vestibular schwannomas” and “vestibular” were selected. The former is considered a diagnosis phrase, whereas the latter is considered an anatomy phrase.

For each selected phrase the annotator searched Wikipedia for the most appropriate page. All three annotators used Wikipedia’s search engine. If a phrase did not have a directly matching Wikipedia page, a more general page was sought that reasonably covers the topic. If no such page was found, the phrase was assigned no Wikipedia page. Thus every phrase was assigned at most one Wikipedia page.

A home-grown annotation tool was used by all three annotators. Upon loading a new report, the tool selected phrases that were selected before by that annotator. The tool also suggested Wikipedia pages for phrases that were annotated before.

The three annotated subsets were merged and consolidated by a single annotator, thus yielding the test collection. In the consolidation phase, the following two properties were ensured:
• When a phrase is selected in one report, it is selected in all reports.

• Two occurrences of the same phrase, possibly in different reports, are assigned the same Wikipedia page, if any.

The second condition says that diagnosis and anatomy phrases are not ambiguous. In generally, this may be a strong assumption. For instance, in the medical domain the word “ventricle” is ambiguous as it may refer to a space in the heart as well as an area in the brain. In our corpus, however, it turned out to be a weak assumption. During the annotation process, no ambiguous phrases were encountered.

In total, 29,256 links, i.e., anchor text–target pairs, are extracted from the 860 reports, which can be resolved to 6,440 unique links. On average, each report contains 34 links.

As our target collection, we use the INEX 2009 Wikipedia collection [219].

9.5.3 Evaluation metrics

We use precision, recall and F-measure as our evaluation metrics. We evaluate the systems’ performance on each radiology report, and show the overall performance which is averaged over all reports. Further, we use a paired t-test for significance testing. A ▲ (▼) indicates a significant increase (decrease) with p-value <0.01; and a △ (▽) indicates a significant increase (decrease) with p-value <0.05.

9.5.4 Preprocessing

We pre-process the Wikipedia collection as well as the radiology reports using Porter stemmer, in order to reduce the morphological variance of terms and phrases. When decomposing anchor texts to sub-anchors, we filter out word sequences that consist of function words only.

9.5.5 Parameter settings

In this section, we specify the parameter settings for each of the automatic link generation systems in our experiments.

Wikify!

We re-implement the Wikify! system as described in [175]. For anchor detection, following [175], we set the threshold \( \tau \) to 6% of the length of the source text. Recall that \( \tau \) is the threshold that selects the top X phrases ranked by the keyphraseness score as anchor texts.
9.5. Experiments

Wikipedia miner

We use the online Wikipedia miner server\(^7\) that is provided by the authors with default parameter settings. The server was accessed remotely and used as a black box.

LiRa

**Anchor detection**  For anchor detection, we use the CRFsuite [185] implementation of CRFs with default parameter settings. For training and evaluating the anchor detection performance, we use 3-fold cross-validation.

As mentioned in Section 9.5.2, the annotations of the anchor texts can overlap. This poses a problem for the sequential labeling approach, as it allows us to assign only one label to each word. For example, in the case of “vestibular schwannomas,” where both “vestibular schwannomas” and “vestibular” are annotated as anchor texts, we have to choose to assign either BOA-EOA or SWA to the word sequence when applying the sequential labeling procedure. In order to solve this problem, we construct the training set with two strategies. With the first strategy, for overlapping annotations, we choose the longer one, and with the second strategy, we choose the shorter one. We refer to the first strategy as *longest labeling (LL)*, and the second as *shortest labeling (SL)*. For example, in the case of “vestibular schwannomas,” in the first setting, we use the label of BOA-EOA for “vestibular schwannomas” and ignore the anchor “vestibular”, and in the second setting, we use the label of SWA for “vestibular” and ignore “vestibular schwannomas.”

**Target candidate identification**  At the target candidate identification stage, we rank Wikipedia pages in descending order of target probability scores and select the top $K$ pages as candidate target pages. Heuristically, we set $K$ to 10.

**Target detection**  We calculate the LLR feature using the first 100 words of each candidate target page. There are two reasons why we select only the first 100 words. First, the first paragraph of a Wikipedia page is usually the summary of the content of that page, and therefore reflects the most important content of that page; the first 100 words is an approximation of the first paragraph of a Wikipedia page. Second, by using a constant number of words from each candidate target page, we eliminate the effect that the total number of words in the page has on the LLR score. This makes the LLR scores comparable across Wikipedia pages.

We experiment with three classifiers: Random Forest (RF) [27], Naive Bayes (NB) and SVM [47], using the Weka implementations [81]. After some preliminary experiments, we found out that RF always outperforms the other two classifiers in terms of both efficiency and effectiveness. Therefore, in the next section, we only focus on the results of RF. To train and evaluate the classifiers, 3-fold cross-validation is used as in the case of training the CRF model for anchor detection.

\(^7\)http://wikipedia-miner.sourceforge.net
Along with the predicted labels, the classifiers also provide a score for prediction confidence. After classification, we execute a post-processing procedure. For anchor texts whose candidate target pages are all classified as “non-target,” we select the candidate target that is predicted as “non-target” with the lowest prediction confidence as the target page. For anchor texts that have multiple candidate target pages classified as “target,” we choose the one with the highest prediction confidence as the target.

9.6 Results

In this section, we show the effectiveness of our proposed approach to automatically generate links from radiology reports to Wikipedia as evaluated on our test collection.

Further, in order to answer RQ7b, How does our proposed approach compare to state-of-the-art approaches that aimed at solving the automatic link generation problem in the general domain?, we conduct a thorough comparison of the performance of our approach to that of the two state-of-the-art systems: Wikify! and Wikipedia miner.

9.6.1 Evaluation on anchor detection

<table>
<thead>
<tr>
<th>System</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRa (LL)</td>
<td>0.90</td>
<td>0.80</td>
<td>0.85</td>
</tr>
<tr>
<td>LiRa (SL)</td>
<td>0.83☆</td>
<td>0.81△</td>
<td>0.82☆</td>
</tr>
<tr>
<td>Wikipedia miner</td>
<td>0.35☆</td>
<td>0.36☆</td>
<td>0.36☆</td>
</tr>
<tr>
<td>Wikify</td>
<td>0.35☆</td>
<td>0.16☆</td>
<td>0.22☆</td>
</tr>
</tbody>
</table>

Table 9.1: Results on anchor detection. LiRa (LL) represents the result of LiRa using longest labeling, and LiRa (SL) represents the result of LiRa using shortest labeling. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa(LL).

Table 9.1 lists the results of anchor detection for the three systems considered in our experiments. Here, two observations can be made. First, LiRa outperforms both Wikipedia miner and Wikify! in anchor detection in terms of all three evaluation metrics, i.e., precision, recall and F-measure. That is, the sequential labeling with CRFs approach trained on radiology data for anchor detection is more effective than the approaches employed by Wikify! and Wikipedia miner, where patterns of anchor texts are learnt from existing Wikipedia links. Second, comparing the two labeling strategies, i.e., LL versus SL, we notice that LL is more effective than SL in terms of precision and F-measure, while SL has a slightly better performance in terms of recall.
9.6. Results

9.6.2 Evaluation on target finding

As discussed in Section 9.5.1, we compare the performance of the three systems in target finding using two sets of anchor texts. Table 9.2 shows the target finding performance of LiRa and Wikify! using annotated anchor texts found in the ground truth. In Table 9.3 we evaluate the performance of LiRa and Wikify! using anchor texts correctly identified by Wikify! and in Table 9.4 we compare the performance of LiRa and Wikipedia miner using the anchor texts correctly identified by Wikipedia miner.

From Table 9.2 and Table 9.3 we see that the target performance of LiRa is better than that of Wikify!. In both cases, i.e., using two different sets of anchor texts, the machine learning based target finding approach of Wikify! is more effective than the Lesk algorithm. Both approaches are less effective, however, than our proposed subanchor-based approach.

Further, from Table 9.4 we see that our proposed approach also outperforms the target finding approach of Wikipedia miner on the same set of anchor texts. The difference in performance between our approach and that of Wikipedia miner is less obvious than that between our approach and the Wikify! system.

<table>
<thead>
<tr>
<th>System</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRa</td>
<td>0.68</td>
<td>0.68</td>
<td>0.68</td>
</tr>
<tr>
<td>Wikify (Lesk)</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>Wikify (ML)</td>
<td>0.26</td>
<td>0.26</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 9.2: Comparing the performance of LiRa and Wikify! on target finding. The target finding algorithms are run on the annotated anchor texts found in the ground truth. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

<table>
<thead>
<tr>
<th>System</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRa</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>Wikify (Lesk)</td>
<td>0.40</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>Wikify (ML)</td>
<td>0.69</td>
<td>0.69</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 9.3: Comparing the performance of LiRa and Wikify! on target finding. The target finding algorithms are run on the anchor texts identified by Wikify!. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

9.6.3 Evaluation on overall system performance

Now we turn to the overall performance of our system in automatically generating links from radiology reports to Wikipedia, compared to the performance of the two state-of-the-art systems.
Table 9.4: Comparing the performance of LiRa and Wikipedia miner on target finding. The target finding algorithms are run on the anchor texts identified by Wikipedia miner. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

<table>
<thead>
<tr>
<th>System</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRa</td>
<td>0.89</td>
<td>0.89</td>
<td>0.89</td>
</tr>
<tr>
<td>Wikipedia miner</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 9.5: Overall system performance. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa(LL).

<table>
<thead>
<tr>
<th>System</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRa (LL)</td>
<td>0.65</td>
<td>0.58</td>
<td>0.61</td>
</tr>
<tr>
<td>LiRa (SL)</td>
<td>0.60</td>
<td>0.58</td>
<td>0.59</td>
</tr>
<tr>
<td>Wikipedia miner</td>
<td>0.29</td>
<td>0.30</td>
<td>0.30</td>
</tr>
<tr>
<td>Wikify! (Lesk)</td>
<td>0.14</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>Wikify! (ML)</td>
<td>0.25</td>
<td>0.12</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Table 9.6: Overall system performance of Wikify! and LiRa in an oracle setting. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

<table>
<thead>
<tr>
<th>System</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRa</td>
<td>0.68</td>
<td>0.68</td>
<td>0.68</td>
</tr>
<tr>
<td>Wikify! (Lesk)</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>Wikify! (ML)</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

In Table 9.5 we show the overall performance of the three systems, which is the final result of anchor detection and target finding. We see that LiRa outperforms the state-of-the-art systems in terms of overall performance, which is within our expectation, since we have already seen that for both anchor detection and target finding, LiRa has shown to be more effective than the other two systems. In addition, for LiRa, if we compare the performance of LL to SL in terms of overall performance, the limited difference in recall as shown in Table 9.1 in anchor detection has disappeared.

In addition, in Table 9.6 we list the overall performance of Wikify! and LiRa using annotated anchor texts found in the ground truth as “recognized anchor texts.” It can be seen as an oracle run of the two systems. That is, if all anchor texts can be correctly identified, we show the performance of the systems on linking these anchor texts to correct target pages in Wikipedia. We see that as in previous experiments, LiRa outperforms Wikify!. However, the performance of LiRa is far from perfect, leaving sufficient room for improvement.
9.6.4 Summary

In this section, we have provided a thorough comparison of our proposed approach to those approaches employed by state-of-the-art systems. With respect to the research question RQ7b, empirical results show that our approach is far more effective than the state-of-the-art approaches in terms of both anchor detection and target finding, and therefore overall performance as well.

In the next section, we further investigate factors that explain the performance difference between systems.

9.7 Discussion

From the description in Section 9.3 we can conclude that a common feature of the two state-of-the-art systems is that they both rely heavily on existing Wikipedia links. While the link structure in Wikipedia has been shown to provide useful training examples for automatic link generation systems in a general domain, it may not be as effective when used as training material for radiology data. As discussed in Section 9.4.1, medical phrases in radiology reports often have a complex semantic structure, for example containing multiple concepts as well as concepts with multiple modifiers. This is intuitively different from existing links in Wikipedia, where the semantic structure of an anchor text is usually less complicated. Or in other words, we expect that the pattern of annotated anchor texts in radiology reports are different from that of the anchor texts found in Wikipedia. Below, we investigate if the difference does indeed exist and whether it has an impact on system performance.

In total, we have 6,440 unique annotated anchor texts in our test collection. In Table 9.7 we list a set of statistics about the coverage of Wikipedia anchor texts over the annotated anchor texts found in our test collection. Let $A^W$ be all the anchor texts found in Wikipedia. We evaluate the coverage on three aspects:

- **exact match** the number of annotated anchor texts occurring in $A^W$;
- **partial match** the number of annotated anchor texts occurring in $A^W$, including the cases when an annotated anchor text is a substring of a Wikipedia anchor;
- **sub exact match** the number of annotated anchor texts containing at least one sub-anchor that occurs in $A^W$.

We see that very few ($<20\%$) annotated anchor texts occur (fully or as a sub string of the anchor texts) in $A^W$. However, over 80\% of the annotated anchor texts do contain one or more concepts, i.e., sub-anchors, occurring in $A^W$.

Now let us look at what these statistics mean to the system performance. For anchor detection, both state-of-the-art systems rely heavily on the Wikipedia anchor texts. The keyphraseness score is used as the only score for identifying anchor texts in the Wikify! system, and used as an important feature for Wikipedia miner. However, from
Table 9.7: The number of annotated anchor texts/sub-anchors in radiology reports covered by Wikipedia anchor texts.

<table>
<thead>
<tr>
<th>Evaluation type</th>
<th>Occur. in WP links</th>
<th>coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact match</td>
<td>923</td>
<td>14.3</td>
</tr>
<tr>
<td>partial match</td>
<td>1,038</td>
<td>16.1</td>
</tr>
<tr>
<td>sub exact match</td>
<td>5,257</td>
<td>81.6</td>
</tr>
</tbody>
</table>

Eq. 9.1, we can see that an anchor text only receives a non-zero score if it occurs in $A^W$. Given the low coverage of the annotated anchor texts in $A^W$, it is not surprising that the keyphraseness score is not effective, as around 85% of the annotated anchor texts would receive a 0 score. LiRa on the other hand, exploits the regularity of the syntactic structure of the annotated anchor texts in the radiology domain. The sequential labeling based approach captures this type of regularity, and is, therefore, effective for anchor detection.

For target detection, all three systems retrieve candidate target pages via Wikipedia links. The difference between the systems can be explained as follows. For Wikify!, candidate target pages are found with respect to an identified anchor text, and for Wikipedia miner, candidate target pages are found with respect to all possible ngrams extracted from a report, while for LiRa, candidate target pages are found with respect to the sub-anchors of an identified anchor text. It is obvious that the approach employed by Wikify! suffers from the same problem as in anchor detection: low coverage of Wikipedia anchor texts over annotated anchor texts in our test collection. LiRa solves this problem using its sub-anchor based approach to retrieve candidate target pages. From Table 9.7, we see that although not perfect, over 80% of the annotated anchor texts have the chance to retrieve their target pages. For Wikipedia miner, although a different strategy is employed, since all possible ngrams in a report are considered, the whole pool of candidate target pages at the report level cover a majority of the annotated target pages for that report. From Table 9.4 we see that this strategy achieves comparable results to our approach.

In summary, we conclude that the reasons why our proposed approach outperforms the state-of-the-art automatic link generation systems are as follows. The low performance of both state-of-the-art systems is mainly due to the complex semantic structure of the annotated anchor texts that are very different from the anchor texts found in Wikipedia. More specifically, the low coverage of Wikipedia anchor texts over the annotated anchor texts in the radiology reports is responsible for the low effectiveness of the two state-of-the-art systems. Our approach caters the complex semantic structure by employing a sub-anchor based approach to target finding and a sequential labeling based approach with syntactic features to anchor detection. Meanwhile the latter effectively exploits the syntactic regularity of medical phrases. Consequently, a much improved result is achieved by our proposed approach.
9.8 Further analysis

In this section, we turn to research question RQ7c: *What is the impact of anchor text frequency on the performance of automatic link generation systems?* More specifically, we investigate: (1) Does the performance of link generation systems show different patterns in recognizing and linking anchor texts with different frequencies of occurrence? (2) Further, if there is a difference, how does it influence the overall performance of the systems?

![Graph of anchor frequency distribution](image)

Figure 9.2: Distribution of anchor frequency. Anchors are ranked according to their frequency of occurrence in the radiology reports. The X-axis shows the logarithm of the ranks of anchors, and the Y-axis shows the logarithm of the frequency of the anchor at that rank.

<table>
<thead>
<tr>
<th>Top 5</th>
<th>Bottom 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass</td>
<td>vestibular nerves</td>
</tr>
<tr>
<td>brain</td>
<td>virchow robins spaces</td>
</tr>
<tr>
<td>meningioma</td>
<td>warthins tumor</td>
</tr>
<tr>
<td>frontal</td>
<td>wegners granulomatosis</td>
</tr>
<tr>
<td>white matter</td>
<td>xanthogranulomas</td>
</tr>
</tbody>
</table>

Table 9.8: (Left) Five most frequent and (Right) five least frequent anchor texts found in the ground truth.

The motivation for the analysis conducted in this section is two-fold. In Figure 9.2, we rank the annotated anchor texts in decreasing order of their frequencies and plot their frequencies with respect to their ranks in the log scale. We see that the anchor text frequencies exhibit typical properties of Zipf’s law [177]. The frequency of an anchor text is inversely proportional to its rank in the frequency table, which forms a distribution consisting of very few words with high frequencies and a long tail of anchor texts with low frequencies. Therefore if the frequency of an anchor text does
Figure 9.3: Systems’ performance differentiated by anchor text frequency. Anchors are ranked according to their frequency of occurrence in the radiology reports. The X-axes show the ranks of anchors, and the Y-axes show the systems’ score on the $r$ most frequent anchors, see Eq. 9.9. Figure 9.3(a) shows the anchor detection rate; Figure 9.3(b) shows the automatic link generation rate. See Table 9.9 for the segmentations of anchor texts based on their frequencies in the test collection.

have an impact on the performance of a link generation system, it is important that a system can correctly recognize and link those rare anchors.

In addition, Table 9.8 shows the five most frequent and five least frequent anchor texts found in our test collection. Intuitively, frequent anchor texts are more likely to be common topics than infrequent anchor texts. By “common topics” we mean that the topic is frequently seen in a general domain and its meaning is more likely to be known to non-experts. For example, in Table 9.8 “brain” is more likely to be a common topic than “xanthogranulomas.” We posit that common topics are more likely to occur in Wikipedia which makes it a relatively easy task for a link generation system, i.e., to identify it as an anchor text and find its target page.

In order to answer RQ7c, we divide the annotated anchor texts into different segments based on their frequencies, as listed in Table 9.9. We then evaluate the performance of the three systems in identifying and finding links for anchor texts in different segments. We evaluate the performance of a system on a segment $seg$ using the following score:

$$
score(seg) = \frac{tp_{seg}}{|seg|},$$

where $tp_{seg}$ is the number of anchor texts within the segment that are correctly recognized in the case of anchor detection, or whose target pages are correctly identified in the case of target finding.

Figure 9.3(a) shows the systems’ performance at anchor detection and Figure 9.3(b) shows the systems’ performance at target finding. Since we do not have access to the intermediate results of the Wikipedia miner system as discussed in Section 9.5.1, here we only show the performance of Wikify! and LiRa on target finding.
9.9 Conclusion

In this chapter, we have studied the problem of automatically generating links from radiology reports to Wikipedia. Two properties set our radiology data apart from data in a general domain, namely, the syntactic regularity and the semantic complexity of the anchor texts, i.e., medical phrases, found in radiology reports. Based on this observation, we proposed an automatic link generation approach for linking medical phrases from radiology reports to concepts in Wikipedia. Using a test collection developed in-house that consists of narrative radiology reports with manually annotated links to Wikipedia pages, we sought answers to three research questions:

RQ7a How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

RQ7b How does our proposed approach compare to state-of-the-art approaches that aimed at solving the automatic link generation problem in the general domain?

RQ7c What is the impact of anchor text frequency on the performance of automatic link generation systems?

Our findings and our answers to the research questions can be summarized as follows.

<table>
<thead>
<tr>
<th>Segments</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq. range</td>
<td>&gt;100</td>
<td>51–100</td>
<td>11–50</td>
<td>6–10</td>
<td>2–5</td>
<td>1</td>
</tr>
<tr>
<td>Num. anchors</td>
<td>116</td>
<td>108</td>
<td>527</td>
<td>482</td>
<td>1399</td>
<td>2149</td>
</tr>
<tr>
<td>Avg. freq.</td>
<td>271.1</td>
<td>70.1</td>
<td>20.7</td>
<td>6.5</td>
<td>2.6</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 9.9: Segmentation of anchor texts based on their frequencies in the test collection.

For both anchor detection and target finding, we see a general trend that better performance is achieved on high frequency anchor texts compared to that on low frequency anchor texts. This observation holds for all systems, which suggests that in general, it may be an easier task for a link system to identify and to find targets for high frequency anchor texts than for low frequent anchor texts. In addition, we see that LiRa shows more robust performance compared to the other systems in that performance remains relatively high even on low frequency anchor texts.

In summary, with respect to research question RQ7c, we have the following answer. We find that anchor frequency has an impact on the performance of link generation systems in both anchor detection and target finding. Empirical results show that in general, link generation systems achieve better performance on high frequency anchor texts than on low frequency anchor texts. Further, since the distribution of anchor frequencies follows Zipf’s law, it is important that a link generation system be effective on low frequency anchor texts, in order to achieve robust performance.
To answer RQ7a, we use a sequential labeling based approach with syntactic features to anchor detection in order to exploit the syntactic regularity present among medical phrases. We then use a sub-anchor based approach to target finding, in order to resolve the complexity in the semantic structure of medical phrases. Our proposed approach has shown to be effective as evaluated on our test collection.

With respect to RQ7b, we find that our proposed approach outperforms two state-of-the-art systems in both anchor detection and target finding, and hence overall performance. Learning the linking patterns from the Wikipedia links, the two state-of-the-art systems failed to capture the domain specific properties of the radiology data, i.e., the syntactic regularity and semantic complexity of the anchor texts in the radiology reports.

Further, with respect to RQ7c, we find that automatic link generation systems tend to achieve better performance in recognizing and finding targets for annotated anchor texts with high frequencies compared to that achieved on anchor texts with low frequencies. Moreover, in order to achieve robust performance, it is important that a system is effective when dealing with low frequency anchor texts.

While our system has shown improved performance over existing automatic link generation systems in the radiology domain, several aspects of the automatic link generation techniques for radiology reports are worth further investigation. For example, in this chapter, we use a purely data-driven approach for both anchor text identification and target finding. An alternative route or extension of the route chosen in this chapter would consider symbolic knowledge representations that are widely available in the medical field, for instance in the form of ontologies. We believe that especially the task of finding a suitable generalization of an anchor text that does not have a matching page in Wikipedia can be achieved by following the hierarchical relationships in an ontology. This research agenda is closely connected to the recent MedlinePlus Connect\textsuperscript{8} activity of the National Library of Medicine in which all SNOMED CT concepts are mapped to pages in Medline Plus.

\textsuperscript{8}http://medlineplus.gov/connect