A wikification prediction model based on the
combination of latent, dyadic and monadic features

Raoni Simões Ferreira

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Raoni Simões Ferreira

A wikification prediction model based on the combination of latent, dyadic and monadic features

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Maria da Graça Campos Pimentel

USP – São Carlos
June 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

F634a	<p>Ferreira, Raoni Simões</p> <p>A wikification prediction model based on the combination of latent, dyadic and monadic features / Raoni Simões Ferreira; orientadora Maria da Graça Campos Pimentel. - São Carlos - SP, 2016. 89 p.</p> <p>Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.</p> <p>1. Wikification. 2. Link prediction. 3. Matrix factorization. 4. Machine learning. 5. Wikipedia. I. Pimentel, Maria da Graça Campos, orient. II. Título.</p>
-------	---

Raoni Simões Ferreira

**Um modelo de previsão para Wikification baseado na
combinação de atributos latentes, diádicos e monádicos**

Tese apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Doutor em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientadora: Profa. Dra. Maria da Graça
Campos Pimentel

USP – São Carlos
Junho de 2016

ACKNOWLEDGEMENTS

Primeiramente, gostaria de agradecer a Deus por ter saúde neste momento tão importante da minha vida e também por permitir que eu esteja sempre rodeado de pessoas maravilhosas e bem intencionados que me apoiam sempre que eu precisei durante minha estadia em São Carlos.

Agradeço à minha orientadora profa Graça Pimentel por ter me aceitado como seu aluno e de oferecer desde do início todo o suporte técnico, financeiro e até psicológico para desenvolver minha pesquisa. Sempre me aconselhando nas dificuldades e motivada a ajudar no que fosse preciso, por ter me oferecido um excelente ambiente de trabalho e proporcionado momentos de descontração, como por exemplo, estendendo a reunião de grupo no Seo Gera :-). Obrigado também por ter sido tão exigente durante esses 5 anos. Você não faz ideia do quanto isso me ajudou e com certeza ajudará na minha carreira. O meu muito obrigado de coração.

Não poderia deixar de registrar o meu agradecimento também ao meu segundo orientador — prof. Marco Cristo que colaborou desde do início com a pesquisa e até o último momento esteve comigo discutindo os resultados e melhor maneira de reportá-los. Marco Cristo acompanha minha vida acadêmica desde quando fui seu aluno de mestrado na Universidade Federal do Amazonas. Foi ele quem me apresentou à profa Graça Pimentel, e por causa desse encontro, surgiu a oportunidade de ingressar no curso de doutorado de uma das melhores universidade do país — a Universidade de São Paulo. O sucesso dessa pesquisa deve-se também às suas valiosas orientações. Não existem palavras pra expressar minha enorme gratidão a ele. Obrigado por ter sido paciente e extremamente exigente. Posso dizer com toda certeza que Marco é uma referência para mim tanto como pesquisador/professor quanto como pessoa. Obrigado meu amigo!

Meus agradecimentos também se estendem aos meus queridos amigos que fiz em São Carlos. Em particular, aos meus amigos do laboratório Intermídia que me acolheram e posso dizer que passaram a ser uma família para mim. Só tenho que agradecer pela amizade que construímos durante esses 5 anos. Em particular e não necessariamente nessa ordem agradeço aos amigos Tiago Trojahn, Johana Rosas, Diogo Pedrosa, Raíza Hanada, Olibário Neto, Andrey Omar, João Paulo, Kleberon Serique, Sandra Rodrigues, Humberto Lidio, Marcio Funes, Kifayat Ullah, John Garavito, Bruna Rodrigues, Flor Karina, Alan Keller, Roberto Rigolin.

Aos meus pais Sidney e Solange e ao meu irmão Rodrigo, que estão em Manaus, por todo amor e apoio incondicional depositado. Eles são minhas referências e tudo que sou deve-se a eles. Sei que estavam na torcida pra eu concluir mais essa etapa na minha carreira. Vocês são demais!!

Não poderia deixar de agradecer também ao amor da minha vida Kamila. Kamila veio para São Carlos em 2012, depois que casamos, para encarar junto comigo essa nova vida morando sozinhos. Ela sabe como foram anos difíceis e ao mesmo de tempo de muito aprendizado o tempo que ficamos aqui em São Carlos. Agradeço também por não ter me abandonado e peço desculpas por ter sido ausente em alguns momentos, principalmente na reta final da tese. Obrigado por todo o seu carinho, amor e dedicação!

Agradeço também ao Governo do Estado do Amazonas por meio Fundação de Amparo à Pesquisa (FAPEAM) pelo apoio financeiro à esta pesquisa.

RESUMO

FERREIRA, R. S.. **A wikification prediction model based on the combination of latent, dyadic and monadic features**. 2016. 89 f. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Atualmente, informações de referência são disponibilizadas através de repositórios de documentos semanticamente ligados, criados de forma colaborativa e com acesso livre na Web. Entre os muitos problemas enfrentados pelos provedores de conteúdo desses repositórios, destaca-se a *Wikification*, isto é, a inclusão de *links* nos artigos desses repositórios. Esses *links* possibilitam a navegação pelos artigos e permitem ao usuário um aprofundamento semântico do conteúdo. A *Wikification* é uma tarefa complexa, uma vez que o crescimento contínuo de tais repositórios resulta em um esforço cada vez maior dos editores. Como consequência, eles têm seu foco desviado da criação de conteúdo, que deveria ser o seu principal objetivo. Isso tem motivado o desenvolvimento de ferramentas de *Wikification* automática que, tradicionalmente, abordam dois problemas distintos: (a) como identificar que palavras (ou frases) em um artigo deveriam ser selecionados como texto de âncora e (b) como determinar para que artigos o *link*, associado ao texto de âncora, deveria apontar. A maioria dos métodos na literatura que abordam esses problemas usam aprendizado de máquina. Eles tentam capturar, através de atributos estatísticos, características dos conceitos e seus *links*. Embora essas estratégias tratem o repositório como um grafo de conceitos, normalmente elas pouco exploram a estrutura topológica do grafo, uma vez que se limitam a descrevê-lo por meio de atributos estatísticos dos *links*, projetados por especialistas humanos. Embora tais métodos sejam eficazes, novos modelos poderiam tirar mais proveito da topologia se a descrevessem por meio de abordagens orientados a dados, tais como a fatoração matricial. De fato, essa abordagem tem sido aplicada com sucesso em outros domínios como recomendação de filmes. Neste trabalho, propomos um modelo de previsão para *Wikification* que combina a força dos previsores tradicionais baseados em atributos estatísticos, projetados por seres humanos, com um componente de previsão latente, que modela a topologia do grafo de conceitos usando fatoração matricial. Ao comparar nosso modelo com o estado-da-arte em *Wikification*, usando uma amostra de artigos Wikipédia, observamos um ganho de até 13% em F1. Além disso, fornecemos uma análise detalhada do desempenho do modelo enfatizando a importância do componente de previsão latente e dos atributos derivados dos *links* entre os conceitos. Também analisamos o impacto de conceitos ambíguos, o que permite concluir que nosso modelo se porta bem mesmo diante de ambiguidade, apesar de não tratar explicitamente este problema. Ainda realizamos um estudo sobre o impacto da seleção das amostras de treino conforme a qualidade dos seus conteúdos, uma informação disponível em alguns repositórios, tais como a Wikipédia. Nós observamos que o treino com documentos de alta qualidade melhora a precisão do método, minimizando o uso de *links* desnecessários.

Palavras-chave: Wikificação, Previsão de links, Fatoração matricial, Aprendizado de máquina, Wikipédia.

ABSTRACT

FERREIRA, R. S.. **A wikification prediction model based on the combination of latent, dyadic and monadic features**. 2016. 89 f. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Most of the reference information, nowadays, is found in repositories of documents semantically linked, created in a collaborative fashion and freely available in the web. Among the many problems faced by content providers in these repositories, one of the most important is Wikification, that is, the placement of links in the articles. These links have to support user navigation and should provide a deeper semantic interpretation of the content. Wikification is a hard task since the continuous growth of such repositories makes it increasingly demanding for editors. As consequence, they have their focus shifted from content creation, which should be their main objective. This has motivated the design of automatic Wikification tools which, traditionally, address two distinct problems: (a) how to identify which words (or phrases) in an article should be selected as anchors and (b) how to determine to which article the link, associated with the anchor, should point. Most of the methods in literature that address these problems are based on machine learning approaches which attempt to capture, through statistical features, characteristics of the concepts and its associations. Although these strategies handle the repository as a graph of concepts, normally they take limited advantage of the topological structure of this graph, as they describe it by means of human-engineered link statistical features. Despite the effectiveness of these machine learning methods, better models should take full advantage of the information topology if they describe it by means of data-oriented approaches such as matrix factorization. This indeed has been successfully done in other domains, such as movie recommendation. In this work, we fill this gap, proposing a wikification prediction model that combines the strengths of traditional predictors based on statistical features with a latent component which models the concept graph topology by means of matrix factorization. By comparing our model with a state-of-the-art wikification method, using a sample of Wikipedia articles, we obtained a gain up to 13% in F1 metric. We also provide a comprehensive analysis of the model performance showing the importance of the latent predictor component and the attributes derived from the associations between the concepts. The study still includes the analysis of the impact of ambiguous concepts, which allows us to conclude the model is resilient to ambiguity, even though does not include any explicitly disambiguation phase. We finally study the impact of selecting training samples from specific content quality classes, an information that is available in some repositories, such as Wikipedia. We empirically shown that the quality of the training samples impact on precision and overlinking, when comparing training performed using random quality samples versus high quality samples.

Key-words: Wikification, Link prediction, Matrix factorization, Machine learning, Wikipedia.

LIST OF FIGURES

Figure 1 – Excerpts of Wikipedia articles <i>Programming Language</i> and <i>Machine Code</i> .	19
Figure 2 – Successive SGD iterations showing the improvement in the estimates while global minimum value is approximated.	31
Figure 3 – Concept graph associated with two example articles Charles Darwin and Stephen Baxter.	44
Figure 4 – Conceptual Architecture for our system of Link Prediction	52
Figure 5 – The Filtering articles module carry out the pre-processing of the original Wikipedia XML file.	58
Figure 6 – WikipediaMiner API was used to extract statistics which summarizes the structure of School collection.	60
Figure 7 – Description of main characteristics of feature concept extraction.	61
Figure 8 – The performance achieved in the test set when used fractions of training set: AUC	66
Figure 9 – The performance achieved in the test set when used fractions of training set: F_1	66
Figure 10 – Proportions of hits and misses, obtained by our complete model, for increasingly ambiguous labels.	70
Figure 11 – Proportions of hits and misses for increasingly ambiguous labels: only disambiguation features	71
Figure 12 – Proportions of hits and misses for increasingly ambiguous labels: only latent component	72

LIST OF TABLES

Table 1 – WikiProject article quality grading scheme	34
Table 2 – Quality Rating Distribution for Wikipedia School and Wikipedia, English Edition	54
Table 3 – Topology statistics of concept graph extracted from Wikipedia School	54
Table 4 – Contingency matrix for link classification	55
Table 5 – Accuracy, AUC, and F_1 figures obtained for classifiers M1, M2, M3, M4, M5, and M6	56
Table 6 – Classifier performance (AUC with 95% confidence intervals) for models composed of a single predictor component C and all components except C , where C is <i>Dyadic</i> , <i>Latent</i> or <i>Monadic</i> . Line <i>All</i> indicates the model composed of all components.	68
Table 7 – Attribute impact when added to latent based model. Confidence intervals are given for a 95% confidence level.	69
Table 8 – Attribute impact when removed from model based on latent and dyadic features. Confidence intervals are given for a 95% confidence level.	69
Table 9 – Performance of anchor classifier according to the quality rate of the training samples measured using AUC, precision and recall with standard error calculated considering 95% confidence levels, along with the corresponding distribution in Wikipedia English Edition and Wikipedia School datasets. . .	74

CONTENTS

1	INTRODUCTION	17
1.1	Context	17
1.2	Motivation	19
1.3	Problem Definition	19
1.4	Research Hypotheses and Questions	20
1.5	Objectives	22
1.6	Contributions	23
1.6.1	<i>Thesis Organization</i>	24
2	BACKGROUND AND RELATED WORK	25
2.1	Notation	25
2.2	Supervised learning	26
2.3	Dyadic prediction problem	27
2.4	Link prediction	27
2.4.1	<i>Existing link prediction models</i>	28
2.4.2	<i>Stochastic Gradient Descent</i>	30
2.5	Wikipedia quality control	32
2.5.1	<i>Discussion on talk pages</i>	33
2.5.2	<i>Content Quality Assessment</i>	33
2.5.3	<i>Linking style</i>	35
2.6	Related work	36
2.6.1	<i>Feature-based wikification</i>	37
2.6.2	<i>Topology-based Wikification</i>	38
2.6.3	<i>Link prediction in other domains</i>	39
2.6.4	<i>Quality of interconnected content</i>	40
2.7	Final Considerations	41
3	A LATENT FEATURE MODEL FOR LINK PREDICTION IN A CONCEPT GRAPH	43
3.0.1	<i>Notation</i>	43
3.1	The Wikification Problem	43
3.2	Wikification Matrix Factor Model	45
3.3	Model Learning	47

3.3.1	<i>Link and Article Attributes</i>	48
3.3.1.1	<i>Link attributes</i>	49
3.3.1.2	<i>Article attributes</i>	51
3.4	Link Prediction System Architecture	51
3.5	Final Considerations	52
4	METHODOLOGY	53
4.1	Wikipedia School Dataset	53
4.2	Evaluation Metrics	54
4.3	Implementation details	57
4.3.1	<i>Filtering articles</i>	57
4.3.2	<i>Obtaining statistics from dump</i>	59
4.3.3	<i>Extracting features from the concept graph</i>	60
4.4	Evaluation Setup	61
4.5	Final Considerations	63
5	EXPERIMENTS AND RESULTS	65
5.1	Comparison with previous models	65
5.2	Analysis of the prediction model components and its attributes	67
5.3	Impact of ambiguity on link prediction	70
5.4	Impact of training samples quality rates on link prediction	72
5.5	Final Considerations	74
6	CONCLUSIONS AND FUTURE WORK	77
6.1	Limitations of this work	78
6.2	Future work	79
6.2.1	<i>Evaluation of the model on different domains and datasets</i>	79
6.2.2	<i>New features</i>	79
6.2.3	<i>Training sample selection based on quality</i>	79
6.2.4	<i>Better understanding of which should be considered appropriate linking</i>	80
6.2.5	<i>Investigate the use of more than one language</i>	80
6.2.6	<i>Adoption of a bipartite ranking approach</i>	80
	BIBLIOGRAPHY	83

INTRODUCTION

After presenting the context and the motivation of our work, in this chapter we define the problem we investigate and present our research hypotheses and questions. Next, we detail our objectives and present a summary of our contributions.

1.1 Context

In the past, the ambition to compile the sum of the human knowledge was the force that drove the creation of many reference works. Unsurprisingly, given the infeasibility of such enterprise, that early dream did not come true. However, as consequence of such efforts, essential knowledge of a range of subjects became accessible to many. Today, even though no reference is expected to be source of all information even about a particular topic, a reference should provide reliable summaries through a network of links which leads on to deeper content of all types — as advocated by Faber (2012).¹

Thus, when readers are faced with a confusing array of resources, modern reference tools provide immediate factual frameworks, context and vocabulary to shape further enquiry, and guidance on where to go next to deepen the understanding. Moreover, Faber (2012) observes that they do that usually on the web, where the questions are asked, and while keeping their complex infra-structure updated. This way, readers can find what they are looking for quickly in a messy and confusing world of knowledge.

Nowadays, reference information is mainly provided through digital libraries comprising large repositories of interconnected articles. Many of these repositories are created collaboratively by voluntary authors and are freely available in the Internet. Among them, the most popular is Wikipedia, a multilingual encyclopedia with over 38 million articles in over 250 different languages. As of February 2014, it had 18 billion page views and nearly 500 million unique

¹ Robert Faber is the Editorial Director of *Oxford Reference* (<<http://www.oxfordreference.com/>>), the general reference platform published by the Oxford University Press.

visitors each month. Only its English version has more than 5 million articles as registered in the [Wikipedia \(2016b\)](#).

One of the main characteristics of Wikipedia is the abundance of links in the body of its articles. The links represent important topical connections among the articles. Such connections provide to readers a deeper understanding of the topics covered by the content they are reading and rich knowledge discovery experiences. [Milne and Witten \(2008\)](#) observed that a common situation faced by users who navigate by Wikipedia is to get lost after following a link associated with an interesting topic that caught their attention and led them to find information they would never searched for — a problem previously studied, among others, by [Conklin \(1987\)](#), [Dillon, Richardson and McKnight \(1990\)](#) and [McAleese \(1989\)](#) while members of the Hypertext community.

As observed by [Adafre and Rijke \(2005\)](#), links in Wikipedia articles are created not only to support navigation but also to provide a semantic interpretation of the content. For instance, links may provide a hierarchical relationship with other articles or a more detailed definition of a concept. Concepts are denoted by anchor texts (source anchors or anchors, for short) which belong to the page content where they are mentioned. They represent the most important elements for the understanding of the article while the links represent the conceptual linkage which semantically approximates the content of different pages. As such, the Wikipedia can be seen as a repository of documents semantically linked which can also be used by software tools associated with various tasks — examples include ontology learning as in the study by [Conde et al. \(2016\)](#), word-sense disambiguation as in the work by [Li, Sun and Datta \(2013\)](#), concept-based document classification as investigated by [Malo et al. \(2011\)](#), and web-based entity ranking as proposed by [Kaptein et al. \(2010\)](#).

The [Wikipedia \(2016a\)](#) regulates that editors are responsible for the identification of anchors and for the selection of the appropriate articles which should be pointed to by the anchors — among other tasks that include the authoring of articles and the verification that Wikipedia authoring guidelines are followed by other editors. The authoring of links is a common practice among editors of reference collections, a task that demands much effort. Given the prominence of the Wikipedia as a collaborative encyclopedia, this task is referred to as *wikification*. To illustrate the wikification task, Figure 1 shows excerpts of two Wikipedia articles. In the first article, *Programming Language*, we observe two sentences. These sentences include six links with anchor texts “formal constructed language”, “instructions”, “machine”, “computer”, “programs”, and “algorithms”. Human editors chose which sequences of words in these sentences should be used as anchors and which links these anchors should point to. In *Programming Language*, for instance, the editor decided that the ambiguous word “instructions” should link to *Machine Code*, thus connecting the concepts *Programming Language* and *Machine Code*. She also defined that words such as “control” and “behavior” would not be anchors while “machine” would appear as anchor only in its first occurrence.

Programming language

From Wikipedia, the free encyclopedia

A **programming language** is a **formal constructed language** designed to communicate **instructions** to a **machine**, particularly a **computer**. Programming languages can be used to create **programs** to control the behavior of a machine or to express **algorithms**.

Machine code

From Wikipedia, the free encyclopedia
(Redirected from **Machine instruction**)

Machine code or **machine language** is a set of **instructions** executed directly by a **computer's central processing unit** (CPU). Each instruction performs a very specific task, such as a **load**, a **jump**, or an **ALU** operation on a unit of data in a **CPU register** or memory. Every program directly executed by a CPU is made up of a series of such instructions.

Figure 1 – Excerpts of Wikipedia articles *Programming Language* and *Machine Code*

Source: Elaborated by the author.

1.2 Motivation

Considering the continuous growth of open reference collections, manual wikification has become increasingly hard. As articles are created and updated, new links have to be added, deleted or updated — a task that would demand editors to be aware of all related topics available in the collection, a requirement hard to meet. As observed by [Huang, Trotman and Geva \(2008\)](#), this also imposes unnecessary efforts on Wikipedia editors who want to act mainly as authors focusing on the content they create and need to keep the creation of links to a minimum. In such a scenario, an automatic tool for wikification would represent an important asset. Also, the understanding of which makes a particular sequence of words an anchor can be useful for many knowledge discovery tasks, such as entity recognition, summarization, and concept representation, to cite a few. From a more ambitious point of view, the research on automatic wikification would represent another step towards the creation of a fully automatic content generator. As it was the case for the automatic creation of links studied in survey by [Wilkinson and Smeaton \(1999\)](#), this has motivated much research on automatic wikification following the seminal work by [Mihalcea and Csomai \(2007\)](#). In general, researchers propose Wikification methods that explore the free availability of Wikipedia content and the large ground truth of encyclopedic links it provides — as in the works contributed by [Mihalcea and Csomai \(2007\)](#), [Milne and Witten \(2008\)](#), [West, Precup and Pineau \(2009\)](#) and [Ratinov et al. \(2011\)](#).

1.3 Problem Definition

Let $\mathcal{A}$ be a set of articles. We define as *label* a sequence of n words, such as “programming” and “programming language”. A title label is associated with each article in $\mathcal{A}$, which we refer to as a *concept*. For instance, the first article in Figure 1 is associated with concept “programming language”. An article can be viewed as a set of labels such as “a”, “programming language”, “language”, “is”, “a”, “formal”, “formal constructed” etc. Each label can be linked to a concept, such as “formal constructed language”. In such case, the label is called an *anchor*.

Given these definitions, the problem of automatic wikification consists in devising an

automatic method able to determine which labels within an article should be linked to concepts available in the collection. Usually, two problems faced by Wikipedia editors, when they decide to place links in an article, have to be addressed:

- How to identify which labels should be anchors. For instance, in Figure 1, the label “machine” was taken as anchor while the label “behavior” was not. We refer to this problem as *anchor detection*;
- How to disambiguate the anchors to the appropriate concepts. For instance, in Figure 1, the label “instructions” was linked to *Machine Code* and not to *Teaching* or *Instruction (a music band from New York)*. We refer to this problem as *link disambiguation*.

In this thesis, we address a relaxed version of these problems, since we view an article as a set of (unique) labels. As such, the reference collection $\mathcal{A}$ can be viewed as a graph where the nodes represent articles and the edges represent the links between the articles. In such a scenario, given two nodes n_1 and n_2 , $n_1 \neq n_2$, the problem is to determine if there should be an edge between n_1 and n_2 . In this thesis, when addressing the problem in this way, we will refer to it as a *link prediction problem*.

1.4 Research Hypotheses and Questions

Wikification methods have used techniques from Machine learning (ML) in general, and from supervised machine learning in particular. Usually the wikification task is modeled as a classification problem where examples of links, described by means of statistical features, are used as training data. In the best methods, these features are designed by human experts and, as result, we refer to them as *human-engineered features*. They capture characteristics of the labels, candidates to be concepts (e.g., their frequency in the article) and its associations (e.g., how related two concepts are). In particular, features about concept associations are very common since wikification can be viewed as the problem of predicting if there should be an edge (link) between two nodes (concepts). The aim is to classify whether an identified concept should be a link to another article. Results reported in the literature have shown the effectiveness of ML-based methods over manual wikification and unsupervised heuristics, as it is the case of the contributions by [Mihalcea and Csomai \(2007\)](#), [Milne and Witten \(2008\)](#), [West, Precup and Pineau \(2009\)](#) and [Ratinov et al. \(2011\)](#).

Despite the fact that the Wikipedia has an underlying graph structure, ML methods currently proposed for wikification exploit the graph topological information taking into account human-engineered features, such as the number of links shared by the articles. As a result, they ignore latent aspects of the graph topology which could be captured by a data-oriented method

such as matrix factorization. This is an important issue since topology information has been successfully used to predict links in many domains using such techniques, as illustrated by the results reported by [Koren \(2008\)](#), [Menon and Elkan \(2011\)](#) and [Rendle \(2012\)](#).

To illustrate, we now consider the movie recommendation domain investigated by [Koren \(2008\)](#) and [Koren \(2009\)](#). In this domain, features should represent very complex user preferences. For instance, a user could prefer horror movies with gore elements, especially if set in space, *or* American science fiction B-movies. These kinds of complex patterns are usual. As the amount of such patterns is huge, it is necessary to determine which ones are the most important and how many of them should be taken so as to capture enough information about all users and movies. To this end, a common strategy is to treat the problem as a rating prediction. Thus, a user-movie rating matrix R has to be approximated by matrix $\hat{R}$ through a *matrix factorization* $\hat{R} = UV$, where U and V are matrices smaller than R . An effective approximation is obtained by the linear combination of k latent features, such that the difference between R and $\hat{R}$ is minimized. The latent features naturally capture how user preferences can be represented in terms of latent aspects that are *relevant in the data*. These aspects would be hardly directly found by human experts because of their complexity.

The optimization approach used in matrix factorization allows for the combination of traditional prediction based on human-engineered features (*side-information*) with prediction based on latent features, as in the work by [Menon and Elkan \(2011\)](#). In the movie recommendation domain, for instance, these combinations are usually adopted to alleviate the cold-start problem, that is, the lack of predictive information about new users and/or items. In such scenario, additional information from the user profile (e.g., demographic data and preferred genres) and from movies (title, year, director) are used as additional predictive evidence.

In sum, given the effectiveness of latent factor approaches in many domains and the limited representation of the topological structure of the concept graph by traditional wikification methods, we can formulate the first hypothesis in this thesis:

Hypothesis 1: *What defines whether two articles are linked to each other is a complex set of criteria. Although these criteria are difficult to be captured by human-engineered features, they may be indirectly captured by the patterns, in the data, which characterize them. Thus, we believe that latent features extracted from the article concept graph might be used to augment the information provided by human-engineered features, resulting in a better prediction model.*

We observe that, by using a data-driven approach, we have to learn the features from the data. Thus, a reliable source of link examples should be available. However, an important issue regarding freely editable reference collections is the quality of the available data. In particular, as we intend to capture link patterns from data, we have to learn from human editors what they consider appropriate links. Given the free and open nature of Wikipedia, the varying background,

expertise, and agenda of the editors, a learner method can be fed with inappropriate examples of links. This is also a relevant because previous studies assume that link information and reference dataset review processes provide a reliable ground truth. However, some studies are cautious about such ground truth and have pointed problems such as persistent missing links, link selection biases and rare use of links, as it is the case in the contributions from [Sunercan and Birturk \(2010\)](#), [Hanada, Cristo and Pimentel \(2013\)](#) and [Paranjape *et al.* \(2016\)](#). To cope with such problems, reference collections such as the Wikipedia adopt mechanisms to assess articles regarding their quality, which includes detailed linking criteria to be checked in review processes.

These observations lead us to our second hypothesis in this thesis:

Hypothesis 2: *Prediction models trained with high quality articles should provide better results in link prediction than models trained with random or lower quality articles.*

Given the previously presented hypotheses, some questions arise:

- How does perform a Wikification model that combines latent- and feature-based prediction components? Do these components provide complementary information such that the combined model is more accurate than its individual components?
- As the latent features represent a reduced version of the original concept graph, do they naturally deal with ambiguity, as different concepts are “located” in different regions in the concept space?
- What is the impact on link prediction of the selection of training samples according to their quality, as assessed by human reviewers?

In this work, we intend to provide answers to such questions. The pursuit for these answers defined our objectives, described in next section.

1.5 Objectives

In this thesis, our main objective is to propose a prediction model to determine which articles should be linked in a reference information repository. The model has to be computationally efficient and effective as solution for the wikification problem. This objective translates into the following specific objectives:

- To propose and evaluate a linear prediction model which combines human-engineered features and latent features. We expect to achieve efficiency by using a linear combination and effectiveness by combining a proven feature-based model with a latent component.
- To determine the importance of each individual prediction component of the model, including the human-engineered features.
- To determine the performance of the method when dealing with ambiguous concepts.
- To determine the effectiveness of the model according to the size of the training dataset and the quality of the training samples selected to compose the training dataset.

1.6 Contributions

In this thesis, we present a novel approach to address the wikification problem that combines the strengths of traditional predictors based on human-engineered statistical features with a latent component which captures the concept graph topology by means of a matrix factorization. This model was evaluated by comparing it with state-of-the-art baselines using training datasets of different sizes, as well as regarding which components and features are the most important, how the model deals with concepts of increasing ambiguity and how the selection of quality training samples have impacted on the learning performance.

The contributions made during this research are summarized as follows:

- A survey on state-of-art approaches to address the anchor prediction and the quality of content in reference repositories and on researches on linking and quality in Wikipedia;
- A scalable and efficient method based on latent features to predict which Wikipedia concepts should be linked;
- A comprehensive evaluation of the proposed method and comparison with state-of-the-art baselines;
- A study on how latent factors could be used for both disambiguation and link prediction;
- A study on how the models trained with different levels of quality impact on wikification.

The first three items above were firstly reported in the paper by [Ferreira, Pimentel and Cristo \(2015\)](#) entitled “Exploring graph topology via matrix factorization to improve wikification” and presented in the 30th ACM Symposium on Applied Computing (SAC 2015). The complete study reported in this thesis was submitted to the Journal of the Association for Information Science and Technology (JASIST) and is currently under review.

To carry out our research, we built an infrastructure in which we implement and evaluate our model; for this we used publicly available code and datasets. Given the importance of those contributions to our work, we will also make publicly available both the code corresponding to our model and the dataset we produced.

1.6.1 Thesis Organization

This thesis is organized as follows. Chapter 2 provides both the background required for understanding the proposal and the related work to this research. Chapter 3 describes the graph prediction problem in the context of link prediction in Wikipedia and the model we proposed to solve the problem. Chapter 4 details the Wikipedia snapshot used and experiments performed to evaluate the model as well as the conceptual architecture we designed to carry out the experiments. Finally, Chapter 5 concludes with perspectives for future work.

BACKGROUND AND RELATED WORK

This chapter describes the mathematical notation used in this thesis and covers background necessary for understanding our approach. Background information includes key concepts and terminologies such as supervised learning, dyadic and link prediction and their relationship with our proposal. In addition, we also describe the quality assessment process adopted by Wikipedia and its associated quality criteria, in particular, regarding the placement of links. Finally, we present an overview of related work highlighting the differences between our proposal and previous work.

2.1 Notation

We use uppercase letters, such as X to denote random variables, boldface uppercase letters, such as $\mathbf{M}$, to denote matrices, and boldface lowercase letters, such as $\mathbf{v}$, to denote vectors. The i th row of $\mathbf{M}$ is denoted as $\mathbf{M}_i$. The element at the i th row and j th column in $\mathbf{M}$ is denoted by M_{ij} . The i th element in $\mathbf{v}$ is denoted by v_i .

In this thesis, we also use coefficients to refer to nodes or pair of nodes. Thus, $\mathbf{v}_i$ and $\mathbf{v}_{ij}$ can denote the vector $\mathbf{v}$ associated with an instance i or a pair of instances (i, j) , respectively. For example, $\mathbf{v}_i$ could represent vector $\mathbf{v}$ associated with article a_i whereas $\mathbf{v}_{ij}$, vector $\mathbf{v}$ associated with the pair of articles a_i and a_j . The same way, we can use Y_{ij} to refer to a scalar associated with pair (i, j) . The intended meaning will be clear given the context.

Given $\mathbf{M}$ and $\mathbf{v}$, we use $\mathbf{M}^\top$ and $\mathbf{v}^\top$ to denote matrix and vector transposes, and $\text{diag}(\mathbf{M})$ to denote matrix $\mathbf{M}$ with entries outside the main diagonal equal zero. The Frobenius norm of $\mathbf{M}$ is given by $\|\mathbf{M}\|_F^2$ and L2-norm of $\mathbf{v}$ by $\|\mathbf{v}\|_2$. Sets are represented by uppercase letters such as $\mathcal{S}$ and its cardinality by $|\mathcal{S}|$.

2.2 Supervised learning

Supervised learning is a machine learning approach which aims to learn patterns in the data based on a set of examples, previously known, denoted as training set. To this end, the learning process tries to construct a mapping function conditioned to the provided training set. Often, learners are tested using unseen data items that compose the so-called test set. The ultimate goal of the learner is to reach some target result with the minimum possible error in unseen data as detailed in the texts by [Faceli *et al.* \(2011\)](#) and [Mitchell \(1997\)](#).

In supervised learning, we have some input space $\mathcal{X} \in \mathbb{R}^D$, a target variable space $\mathcal{Y} \subseteq \mathbb{R}$, and a training set comprising n samples of the form $\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ from some distribution $P(X, Y)$ over $X \times Y$, where X, Y are random variables associated with domains $\mathcal{X}$ and $\mathcal{Y}$. In each pair $(\mathbf{x}_i, y_i)$, vector $\mathbf{x}_i \in \mathcal{T}$ denotes one or more features (attributes), i.e., statistic values that represent measurement likely correlated to target result $y_i \in \mathcal{Y}$. For instance, if $\mathbf{x}_i$ is a pair of Wikipedia articles, a possible feature could be how many incoming links the articles share. The learning method attempts to combine these features to predict the value of y_i . More formally, given a training set $\mathcal{T}$, the aim is to learn a function $f: \mathcal{T} \rightarrow \mathcal{Y}$ that generalizes well with respect to some loss $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$, i.e:

$$\mathbb{E}[f]_{(\mathbf{x}, y) \in \mathcal{T}} \sim \ell(f(\mathbf{x}), y) \quad (2.1)$$

for which we expect error/risk $\mathbb{E}[f]$ is small enough. We can minimize $\mathbb{E}$ by minimizing its empirical counterpart,

$$\hat{\mathbb{E}}[f] = \frac{1}{n} \sum_{k=1}^n \ell(f(\mathbf{x}_k), y_k) \quad (2.2)$$

When the mapping f is parametrized by some $\theta \in \Theta$, we have to solve the following optimization problem:

$$\underset{\theta \in \Theta}{\text{minimize}} \quad \frac{1}{n} \sum_{k=1}^n \ell(f(\mathbf{x}_k; \theta), y_k) \quad (2.3)$$

Once this problem is solved, we can evaluate how good our estimate is. Several mathematical techniques were proposed to solve this optimization problem. In [Section 2.4.1](#) we detail a technique used in this work.

When the target value we intend to predict (y_i) can take only a small number of discrete values or classes (e.g predict if a pair of articles should be linked or not), we refer to the prediction as *classification*. When the target value can assume any value in a real interval, the prediction is called *regression*. In this work, despite we address wikification as a classification problem (two concepts are linked or not linked), we adopt a solution based on regression. In particular, we

take the real numbers predicted by a regressor as estimates of how likely the items belong to particular classes (cf. Chapter 3 for details).

2.3 Dyadic prediction problem

Dyadic prediction is a generalization of problems whose main objective is to predict the interactions between pair of objects as observed by Menon (2013). In dyadic prediction, the training set consists of pairs of objects $\mathcal{T} = \{(i, j)\}_{i=1}^n$ called dyads, with associated target variable $y_{i=1}^n$. The information available about each dyad member is at least a unique identifier, and possibly some additional features or *side-information*. The task is to predict targets for unobserved dyads, that is for pairs (i', j') that do not appear in training set. Note that, dyadic prediction problem can be defined as a special case of supervised learning as detailed by Menon (2013) and Tsoumakas and Katakis (2007).

Dyadic prediction is defined as follow. Let $\mathcal{I}, \mathcal{J} \subseteq \mathbb{Z}_+$ be two sets of non-negative integers, $\mathcal{A} = \mathcal{I} \times \mathcal{J}$ be the set of pairs of these integers, and $\mathcal{Z}_1 \subseteq \mathbb{R}^{D_1}, \mathcal{Z}_2 \subseteq \mathbb{R}^{D_2}$. Further, let $\mathcal{X} \subseteq \mathcal{A} \times \mathcal{Z}_1 \times \mathcal{Z}_2$, and $\mathcal{Y}$ be some arbitrary set. Set $\mathcal{A}$ defines all possible dyads by means of positive integer identifiers, with $\mathcal{I}$ and $\mathcal{J}$ being the sets of identifiers for the individual dyad members i and j respectively. Sets $\mathcal{Z}_1, \mathcal{Z}_2$ denote the *side-information* for the individual dyad member and the dyad pair, respectively. Set $\mathcal{X}$ consists of the union of the dyads and their side-information, while $\mathcal{Y}$ represents the target variable space. The problem of dyadic prediction is to learn a function $\hat{Y}: \mathcal{T} \rightarrow \mathcal{Y}$ that generalizes well with respect to some loss function $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ where $\hat{Y}$ returns an estimate for a given dyad, and the loss ℓ measures how good this estimate is.

A typical approach to deal with prediction for unobserved dyads is to address the dyadic problem as being one of filling the entries of an incompletely observed matrix $\mathbf{M} \in \mathcal{Y}^{|\mathcal{I}| \times |\mathcal{J}|}$, where $i \in \mathcal{I}, j \in \mathcal{J}, y_{ij} \in \mathcal{Y}$. This approach is referred to as *matrix completion* in the contributions by Koren (2008), Koren (2009), Menon and Elkan (2010) and Menon and Elkan (2011). Each row of $\mathbf{M}$ is associated with some $i \in \mathcal{I}$ and each column with some $j \in \mathcal{J}$, so the training data is a subset of observed entries in $\mathbf{M}$. In this work, we adopt the same approach (cf. Chapter 3).

The dyadic prediction encompasses many real world problems where the input is naturally modeled as interaction between objects. Important cases include movie recommendation, predicting links in social network, predicting protein-protein interaction and so on.

2.4 Link prediction

As observed by Menon (2013), link prediction is a special case of dyadic prediction problem concerned with problems related to prediction of presence or absence of edges between nodes of

a graph. There are two main research directions in link prediction based on the dynamism of the analysed network. One research direction is *structural link prediction*, where the input is a partially observed graph, and we wish to predict the status of edges for unobserved pairs of nodes as it is the case of the contributions by [Adamic and Adar \(2003\)](#), [Newman \(2001\)](#) and [Menon and Elkan \(2011\)](#). Other research direction is *temporal link prediction*, where we have a sequence of fully observed graphs at various time steps as input, and our goal is to predict the graph state at the next time step; examples include the studies by [Liben-Nowell and Kleinberg \(2007\)](#), [Dunlavy, Kolda and Acar \(2011\)](#) and [Rümmele, Ichise and Werthner \(2015\)](#). Note that, unlike the latter, in structural link prediction, the temporal aspect is omitted because its focus is on the analysis of a single snapshot of the graph. Both research directions have showed important findings in practical applications, such as predicting interactions between pairs of proteins and recommending friends in social networks respectively.

In this thesis, we are interested in structural link prediction on directed graphs which possess only a single edge between a pair of nodes (i.e, adjacency matrix $\mathcal{M}$ is binary). Thus, hereinafter, we focus on such graphs. Formally, training set $\mathcal{O}$ consists of a partially observed graph $\mathbf{G} \in \{0, 1, ?\}^{n \times n}$, where 0 denotes a *known absent* link, 1 denotes a *known present* link, and ? denotes an unknown status link. The set of observed dyads is denoted by $\mathcal{O} = \{(i, j) : G_{ij} \neq ?\}$. Unknown ? means that, for some pairs of nodes (i, j) , we do not know whether or not there exists an edge between them. The goal is to make predictions for all such node pairs entries in $\mathbf{G}$. Because this is a dyadic problem, we may have a feature vector associated with pairs of nodes (i, j) (denoted as $\mathbf{z}_{ij}$) and with each individual node (denoted as $\mathbf{x}_i$ and $\mathbf{x}_j$).

Next, we will present the main approaches to dealing with the link prediction problem.

2.4.1 Existing link prediction models

Existing prediction models are categorized into two main classes: unsupervised and supervised. Unsupervised approaches are based on certain topological properties of the graph such as the degree of the nodes, the set of neighbours the nodes share or based on some scoring rules such as the proposed by [Adamic and Adar \(2003\)](#), the Katz score contributed by [Lü and Zhou \(2011\)](#) and similarity scores (e.g., co-occurrence and Pearson correlation). These scores serve as indicators of the linking likelihood between any pair of nodes. These models tend to be rigid as they use predefined scores that are invariant to the specific structure of the input graph and thus do not involve any learning as observed by [Menon and Elkan \(2011\)](#). Despite this limitation, they demonstrated very successful results, for instance, in prediction of the collaborations in co-authorship networks, as in the results reported by [Liben-Nowell and Kleinberg \(2007\)](#), and on the earliest collaborative filtering approaches contributed by [Resnick et al. \(1994\)](#). For more detail about them, we refer the reader to surveys by [Liben-Nowell and Kleinberg \(2007\)](#) and [Adomavicius and Tuzhilin \(2005\)](#).

On the other hand, the supervised approaches directly capture the link behaviour from a

set of observed samples. The idea behind these models is that an estimate about the existence of link between two nodes is a function among a series of statistics about the nodes and its associations leveraged by a set of associated parameters, given by vector Θ . Thus, the method learns vector Θ from the observable samples by minimizing the error between the estimate and the real values. In general, such models can be described by Equation 2.4:

$$\underset{\Theta}{\text{minimize}} \frac{1}{|\mathcal{O}|} \sum_{(i,j) \in \mathcal{O}} \ell(\hat{G}_{ij}(\Theta), G_{ij}) + \Omega(\Theta) \quad (2.4)$$

where $\hat{G}_{ij}$ is the model prediction for dyad (i, j) , $\ell(\cdot, \cdot)$ is a loss function, $\Omega(\cdot)$ is a regularization term that prevents overfitting and $\mathcal{O} = (i, j) : G_{ij} \neq ?$ is the set of observed dyads. The choice of these terms depends on the type of the model. In this work, we focus on models well studied on literature, namely the feature-based and the latent-based modes.

- **Feature-based model.** A typical feature-based model ignores the dyad members' identifiers and applies supervised learning on the features. The model assumes that each node i in the graph has an associated feature vector $\mathbf{x}_i \in \mathbb{R}^d$ and each edge (i, j) has a feature vector $\mathbf{z}_{ij} \in \mathbb{R}^D$. $\hat{G}_{ij}(\Theta)$, in Equation 2.4, can be instantiated:

$$\hat{G}_{ij}(\Theta) = L(f_E(\mathbf{z}_{ij}; \mathbf{w}) + f_N(\mathbf{x}_i, \mathbf{x}_j; \mathbf{v}_i, \mathbf{v}_j)) \quad (2.5)$$

for appropriate scoring functions $f_E(\cdot)$, $f_N(\cdot)$ acting on edges (the dyadic features) and on nodes (the monadic features), respectively, and a link function $L(\cdot)$. Generally, the adopted link function normalizes the estimate such that $\hat{G}_{ij} = 1$ if (i, j) is a link and 0, otherwise. Weight vectors $\mathbf{v}_i$, $\mathbf{v}_j$, and $\mathbf{w}$ are associated with feature vectors $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{z}_{ij}$, respectively. A common choice for edge and node score functions is a linear combination of $f_E(\mathbf{z}_{ij}; \mathbf{w}) = \mathbf{w}^\top \mathbf{z}_{ij}$ and $f_N(\mathbf{x}_i, \mathbf{x}_j; \mathbf{v}_i, \mathbf{v}_j) = \mathbf{v}_i^\top \mathbf{x}_i + \mathbf{v}_j^\top \mathbf{x}_j$. Note that, as defined, f_N does not properly capture affinities between nodes i and j that could be observed through the features $\mathbf{x}_i$ and $\mathbf{x}_j$. Thus, a better choice for this function would be a bilinear regression defined as $f_N(\mathbf{x}_i, \mathbf{x}_j; \mathbf{v}_i, \mathbf{v}_j) = \mathbf{x}_i^\top \mathbf{V} \mathbf{x}_j$, where $\mathbf{V} \in \mathbb{R}^{n \times n}$ (n is the number of nodes) is a set of weights between each possible pair of nodes.

Although this model captures characteristics of a graph $\mathbf{G}$, it takes limited advantage of the topological structure of $\mathbf{G}$. Recently, link prediction models have been used to exploit both dyad member identifiers (i and j) and feature score functions f_E and f_N , when available. They describe $\mathbf{G}$ by means of latent features extracted from dyad members using data-oriented approaches such as Matrix Factorization, which is described in the following paragraph.

- **Latent-based model (Matrix factorization).** This model has provided state-of-the-art performance in link prediction and other dyadic prediction problems, and is the main

focus of this work. The idea is to learn latent features through matrix factorization. The link prediction task is addressed as a matrix completion problem where $\mathbf{G}$ is factorized as $L(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top)$ for some $\mathbf{U} \in \mathbb{R}^{n \times k}$, $\mathbf{\Lambda} \in \mathbb{R}^{k \times k}$ and a link function $L(\cdot)$. Each node i has a corresponding latent vector $\mathbf{u}_i \in \mathbb{R}^k$, where k is the number of latent features. After instantiation of $\hat{G}_{ij}$, Equation 2.4 translates into:

$$\underset{\Theta}{\text{minimize}} \quad \frac{1}{|\mathcal{O}|} \sum_{(i,j) \in \mathcal{O}} \ell(L(\mathbf{u}_i^\top \mathbf{\Lambda} \mathbf{u}_j), G_{ij}) + \Omega(\mathbf{U}, \mathbf{\Lambda}) \quad (2.6)$$

where regularizer $\Omega(\mathbf{U}, \mathbf{\Lambda}) = \frac{\lambda}{2} \|\mathbf{U}\|_2 + \frac{\lambda}{2} \|\mathbf{\Lambda}\|_2$.

Note that the best approximation that agrees with $\mathbf{G}$ is derived from only the observed entries $\mathcal{O}$. The hope is that with suitable priors on $\mathbf{U}$ and $\mathbf{\Lambda}$ the approximation will generalize to the missing entries in $\mathbf{G}$ as well, as argued by [Menon and Elkan \(2010\)](#).

2.4.2 Stochastic Gradient Descent

As previously mentioned in Section 2.2, the supervised learning approach for link prediction attempts to learn the model parameter vector Θ . To this end, we minimize the sum of the regularized error. An easy and popular way to solve this sum is by using *stochastic gradient descent* optimization (SGD) reviewed by [Bottou \(2012\)](#). SGD approximates the error of the loss function by simultaneously updating all the weights Θ of the linear estimate, according to its gradients. Suppose we intend to minimize a loss function that has the form of a sum:

$$\varepsilon(\Theta) = \sum_i \ell_i(\Theta) \quad (2.7)$$

where the parameters Θ which minimize $\varepsilon(\Theta)$ should be estimated.

SGD will use information about the slope of function $\varepsilon(\Theta)$ to find a solution Θ for which $\varepsilon(\Theta)$ is a global minimum. It starts with usually a random estimate Θ_t , $t = 0$, and then checks how small is the observed loss $\ell(\Theta_t)$ for m samples taken at random (thus the name *stochastic*)¹. If the difference between successive estimates is not small enough, current estimate Θ_t is improved with a small step, proportional to the gradient of $\ell(\Theta_t)$, i.e., $\Theta_t = \Theta_t - \gamma \nabla \ell(\Theta_{t-1})$, where γ defines the size of the step (the *learning rate*) and $\nabla \ell(\Theta_{t-1})$ is the gradient of $\ell(\Theta_{t-1})$. Figure 2 illustrates this algorithm.

The selection of an appropriate learning rate γ is very important because if γ decreases with an appropriate rate, SGD converges almost surely to a global minimum when the objective function is convex or pseudoconvex, and otherwise converges almost surely to a local minimum, as observed in the review by [Bottou \(2012\)](#).

¹ Normally, in SGD, the gradient is approximated using a single example, i.e., $m = 1$. This is the case in this thesis. SGD is a variation of the method called “Gradient Descent”, for which $m = n$, where n is the number of training examples. Another popular variation is “Mini-batch Gradient Descent”, where $1 < m \ll n$.

Figure 2 – Successive SGD iterations showing the improvement in the estimates while global minimum value is approximated. Functions $\ell(\Theta)$ and $\nabla\ell(\Theta)$ are shown in blue and brown, respectively. Assuming $\gamma = 1$, SGD starts with Θ_0 and then moves to the right with step size $\gamma \nabla\ell(\Theta_0)$. As the new point found, Θ_1 , is greater than the global minimum, the new step, $\gamma \nabla\ell(\Theta_1)$, is positive and moves SGD leftwards, to point Θ_2 . At each iteration t , Θ_t is closer to the global minimum. Note that the less steep is the curve, the smaller is the slope (gradient) and, by extension, the step.

Source: Elaborated by the author.

In our scenario, to determine how weights are updated we derive the regularized loss function in Equation 2.6 with respect to Θ to obtain its gradient. By assuming $\Theta = (\mathbf{U}, \mathbf{\Lambda})$ and a undirect underlying graph (for simplicity), we obtain the following weight update expressions:

$$\begin{aligned} \mathbf{U}_i &= \mathbf{U}_i - \gamma ((L(\hat{G}_{ij}(\Theta)) - G_{ij})) \mathbf{\Lambda} \mathbf{U}_j + \lambda \mathbf{U}_i \\ \mathbf{U}_j &= \mathbf{U}_j - \gamma ((L(\hat{G}_{ij}(\Theta)) - G_{ij})) \mathbf{\Lambda}^\top \mathbf{U}_i + \lambda \mathbf{U}_j \\ \mathbf{\Lambda} &= \mathbf{\Lambda} - \gamma ((L(\hat{G}_{ij}(\Theta)) - G_{ij})) \mathbf{U}_i \mathbf{U}_j^\top + \lambda \mathbf{\Lambda} \end{aligned} \quad (2.8)$$

where (i, j) represents a pair of nodes (not necessarily linked), γ is the learning rate, and λ is a parameter which controls how important the regularization is. In practice, for each pair (i, j) (with known status G_{ij}), a prediction $\hat{G}_{ij}$ is made, and the associated prediction error is computed. Thus, for each pair (i, j) we modify the parameters by moving in the opposite direction of the gradient. The complete algorithm, applied to our scenario, is presented in Chapter 3.

2.5 Wikipedia quality control

Wikipedia is the world's largest online encyclopedia in terms of size, scope, and availability. It comprises more than 38 million articles in more than 250 languages accessed by about 500 million unique visitors each month as registered in [Wikipedia \(2016c\)](#). Since it was released in 2001, Wikipedia has adopted the model of open editing policy which encourages anyone make contributions to its articles. This vision enabled an unprecedented growth in terms of size and coverage of the content when compared to traditional encyclopedias. For instance, the English edition of Wikipedia reached two million articles on 2007, making it the largest encyclopedia ever assembled, surpassing the Chinese Yongle Encyclopedia, which had held the record for almost 600 years as informed by [Wikipedia \(2016b\)](#).

All people who contribute to Wikipedia are volunteer wikipedians or editors² and they are usually motivated to contribute with subjects that they have personal interest or familiarity. By 2016, Wikipedia counted with about 27 million registered editors according to [Wikipedia \(2016b\)](#). Basically, these editors can act both as authors and reviewers. Reviewing an article includes several tasks, such as fixing typos, deleting inadequate updates (e.g attempts of vandalism), resolving disputes, and perfecting content. As every edition is recorded, it can be reverted by any other editor. Each version of an article is available in its revision history and can be compared to other versions.

Although Wikipedia offers an environment of free editions, these activities are ruled by some principles of etiquette and editing policies or guidelines to preserve the healthy relationship between editors. Principles of etiquette reflect conducts widely acceptable between editors and should be followed when they work in collaboration. Examples of such principles include the respect and civility even when they have different points of view, the avoidance of update reversions without previous discussion, no engagement on personal attacks and edit wars etc³. On the other hand, editing policies refer to the best practices that the editors should follow such as verifiability (any claim should be well supported by solid reference), to keep a neutral point of view (relying on multiples sources instead of only promoting a certain view), not to add original research, and to avoid using Wikipedia as a discussion forum⁴.

In general, Wikipedia gets more well-intentioned editors, which respect its principles and policies than bad ones. And even when bad editors edit articles, they rarely get support and their edits are rapidly reversed. Therefore, the collaborative control of content quality of Wikipedia is composed mostly by well-intentioned editors who regularly and constantly watch over articles. For instance, it is usual that editors track articles of their interest that were recently modified, mark articles with pending issues to solve later, and discuss improvements to an article by joining a talk page as suggested in [Wikipedia \(2016a\)](#).

² [<https://en.wikipedia.org/wiki/Wikipedia:Wikipedians>](https://en.wikipedia.org/wiki/Wikipedia:Wikipedians)

³ [<https://en.wikipedia.org/wiki/Wikipedia:Etiquette>](https://en.wikipedia.org/wiki/Wikipedia:Etiquette)

⁴ [<https://en.wikipedia.org/wiki/Wikipedia:Editing_policy>](https://en.wikipedia.org/wiki/Wikipedia:Editing_policy)

Wikipedia, however, does not rely only on this informal control of quality. It also adopts a systematic quality control policy that encompasses a quality assessment process. In the next sections, we detail some aspects of Wikipedia that are important to this work.

2.5.1 Discussion on talk pages

With each article in Wikipedia, there is an associated talk page. They provide space for editors discuss changes made to improve the associated articles. Although the editors can simply edit, they are encouraged to express their concerns (e.g., something they do not totally agree with), get feedback or help other editors working on the article. Besides, they can mention possible problems, leave notes about current or ongoing work on the article, and negotiate solutions for conflicts. They play an important role in Wikipedia because editors can discuss the content without leaving comments in the actual article itself as observed by [Ayers, Matthews and Yates \(2008\)](#).

Talk pages are also the place where it is possible to find essential information about the articles, such as their quality rating, the categories they belong, their importance, their versions in other languages, the archive links to older talk pages discussion, and so on. In particular, quality ratings are obtained through the Wikipedia content quality assessment process described in next section.

2.5.2 Content Quality Assessment

Wikipedia articles are in constant state of development with editors making contributions by adding new articles or improving existing ones. Content quality varies widely from article to article. While most of them are useful as a basic reference, many are still incomplete. Some articles are unreliable to the point that some caution is advisable to whom read them.

[Ayers, Matthews and Yates \(2008\)](#) argue that, whereas to readers it could be hard to judge the value of information they are looking at, editors must be able to discern what could be improved about an article when they work on it. To assist editors on such task, the Wikipedia community has established a formal review process which includes article quality assessments and associated manuals of style guidelines.⁵ Through this process, quality rates can be assigned to articles.

In particular, the quality rating system is based on a quality scale, where letters are used to indicate the quality of the article. It reflects mainly how factually complete the article is about a particular topic considering, for example, its content, its structure and quality of writing. This system also serves as an indicative, to Wikiprojects, of the current level of contribution of the article. Wikipedia adopts a quality scale of seven levels, although, in the research literature, it is common the reference to an earlier six-level scale as it is the case in the contributions

⁵ https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking

by Ayers, Matthews and Yates (2008) and Dalip *et al.* (2011). We here adopt the current quality scale, defined in the Wikipedia⁶ and summarized in Table 1.

Table 1 – WikiProject article quality grading scheme

Class	Criteria
Featured Article (FA)	These are articles which exemplify the very best work according to their evaluators. No further content additions should be necessary unless new information becomes available; further improvements to the prose quality are often possible.
A-Class (AC)	These are articles essentially complete about a particular topic but they still have a few pending issues that need to be solved in order to be promoted to FA. They are well-written, clear, complete, have appropriate length, structure, and reference reliable sources. They include illustrations with no copyright problems. After addressed possible minor style issues, they can be submitted to a peer-reviewed featured article evaluation.
Good Article (GA)	Articles without problems of gaps or excessive content. They are considered good sources of information, although other encyclopedias could provide better content. Articles with this rate should comply with the manual of style guidelines for most items, including linking.
B-Class (BC)	Articles that are useful for most users, but researchers may have difficulties in obtaining more precise information. Starting on this rating, article should be checked for general compliance with the Manual of Style and related style guidelines, including linking guidelines.
C-Class (CC)	Articles still useful for most users, but which contain serious problems of gaps and excessive content. Such articles would hardly provide a complete picture for even a moderately detailed study.
Start-Class (ST)	Articles still incomplete although containing references and pointers for more complete information. It may not cite reliable sources.
Stub-Class (SB)	These are draft articles with very few paragraphs. They also have few or no citations. They provide very little meaningful content with insufficiently developed features of the topic.

In general, any user can assign a rating SB, ST, CC, and BC to an article. To assess an AC rating, it is necessary the agreement of at least two editors. Ratings GA and FA should only be used on articles that have been reviewed by a committee of editors, known as a WikiProject. Wikiprojects are composed by groups of editors/reviewers who are specialized in certain topics like Biology, Science, or History. They bear ultimate responsibility for resolving disputes.

It is important to highlight that an article can be rated by multiple Wikiprojects. Thus, the same article can be rated with distinct quality labels because each wikiproject can adopt

⁶ <https://en.wikipedia.org/wiki/Wikipedia:Version_1.0_Editorial_Team/Assessment>

different criteria. Since we do not deal with multiple ratings, we take the article rating the one given by the largest number of Wikiprojects. In case of tie, we choose one arbitrarily.

As previously mentioned, the review and quality assessment process is guided by detailed manuals of style. The aim of such guides is to promote clarity and cohesion in writing as well as assistance on language use, layout, and formatting. Such guides cover a large number of specific topics such as punctuation, structural organization, use of capital letters, ligatures, abbreviations, italics, quotations, dates and time, numbers, currency, units of measurement, mathematical symbols, grammar and usage, vocabulary, images, numbered lists etc. Among all these topics, the most important for this thesis is linking. We present a summarized description of Wikipedia linking guidelines in next section.

2.5.3 *Linking style*

As a hypertext-based repository, linking is an important feature of Wikipedia. Linking guidelines cover diverse topics such as placement techniques, maintenance, exceptions, and degree of specificity, to cite a few. Among them, we are more interested in suggestions about which concepts should be linked.

Concerning this topic, Wikipedia recommends that links should be placed where they are relevant and helpful in the context. In general, concepts should be linked when the target article:

- Will help the reader to understand the topic more fully. This is particularly common in certain locations of the articles, such as the article leads (the introduction of an article), the opening of new sections, table cells, infobox fields, and image captions, to cite a few;
- Provides background information;
- Explains technical terms, jargon, slangs etc. Note that articles about technical topics are expected to be more densely linked as they probably contain more technical terms;
- Provides additional information about a proper name that is likely unfamiliar to the reader.

On the other hand, editors are advised to avoid the excessive use of links, a problem referred to as *overlinking*. This is a serious issue because excessive links can distract the reader and difficult future maintenance. Thus, the guide suggests the editors avoid placing links in the following situations:

- The concept is generally understood by most readers, such as everyday words, common occupations, common units of measurement, dates, names of major geographic features and locations, languages, nationalities, and religions;

- The term to be linked was already linked in the same article. Note, however, that repeated links are recommended in infoboxes, captions, footnotes etc, as a matter of convenience;
- The concept occurs in certain article locations such as section headings, the opening sentence of a lead, within quotations, and immediately after another link so that they would look like a single link;
- The concept could be explained with very few words making the link somewhat unnecessary. In general, it is recommended to avoid forcing the user to follow a link to understand a sentence or term. Such encouraging of self-sufficiency leads to content suitable to situations where link navigation is undesired (e.g., mobile environments) or not possible (e.g., a print copy of the content);
- If the main intention behind the link is to draw attention to certain words or ideas. As previously mentioned, links should be used to provide deeper understanding and not to distract the reader;

Among placement recommendations, the most important is to link to the correct concept, which implies concept disambiguation. For instance, in a page about supply and demand, the editor should link “good (economics)” and not “good”. Other recommendations include clarity (e.g., in sentence “When Mozart wrote his Requiem”, the editor should choose as anchor text “his Requiem” instead of “Requiem”), specificity (e.g., in previous sentence, she should link to “Requiem (Mozart)” instead of only “Requiem” or the combination “Requiem” and “Mozart”), redirections (e.g., in sentence “She owned a poodle”, “poodle” would redirect to “Dog”), and specific cases such as when one should link to dates or measurement units.

All previous recommendations are not only important because they guide the editors, but also because they are used by the reviewers to verify if editors place links appropriately. Thus, by following these suggestions, editors and reviewers adopt standard criteria that make it possible the proposal of automatic methods based on machine learning, such as the one described in this thesis.

2.6 Related work

In this section, we review the research related to the wikification problem. We start by prediction methods based on machine learning. We refer to these methods as feature- and topology-based. We then highlight some results which have motivated the ML community to use latent feature models for link prediction, a related problem to wikification. Finally, we show related studies on automatic quality assessment and on quality and linking.

2.6.1 Feature-based wikification

In the feature-based wikification approach, the objective is usually to develop learning models to recommend links to newly created Wikipedia articles. Such models have to learn what should be considered an appropriate link according to human editors. Thus, link patterns are learned from existing articles, available in the collection. They assume the input data are the raw content of the articles (without outgoing links) or any other textual document. The system normally has to perform two different predictions (i.e. classifications):

- *Anchor detection*: identification of which term or sequence of terms (which we refer from now on as *label*) should be an anchor;
- *Concept disambiguation*: identification of the article to which the link associated with the anchor should point to.

An illustration of the approach is as follows: suppose an author is editing an article about “Programming Languages” which reads “Newer programming languages like Java and C# have definite assignment analysis”. In this sentence, the anchor detection should select labels such as “Java”, “C#”, and “definite assignment analysis” to be anchors. Regarding the anchor “Java”, the concept disambiguation has to place a link to point to the concept “Java (Programming Language)” and not to alternatives also referred to by label “Java”, such as “Java (Island)”, “Java (Indonesian Language)”, “Java (coffee)” etc.

To the best of our knowledge, [Mihalcea and Csomai \(2007\)](#) were the first to address these problems using this ML approach and also introduce the use of Wikipedia as a source for develop concept extraction and word sense disambiguation. They first identified the probability of a term to be used as anchor according to a probability threshold. They disambiguate terms by using a supervised classifier based on features such as context textual clues and part-of-speech tags. Their findings show that the results provides by an automatic wikification system were similar those provides by manual system.

Soon afterwards, [Milne and Witten \(2008\)](#) proposed an alternative approach where the disambiguation task should precede the anchor identification. Both tasks were based on supervised techniques. For disambiguation, a classifier was trained using features such as how often an article is used in the correct sense, how related are the source and target articles, and the quality of the context. For identification, another supervised classifier was used which took into consideration statistics from concepts such as its position in the text.

Leveraging up the aforementioned contributions, [Ratinov et al. \(2011\)](#) proposed a ML-method that combines text content features (i.e local features) and link graph features (i.e global features) from Wikipedia articles so as to link named entities to Wikipedia concepts found in

general text. Their results applied in news domain surpassed previous approaches although methods based on local features still offers competitive results.

As a follow up of their previous work, [Milne and Witten \(2013\)](#) extended their approach so as to integrate a set of mining tools specifically built to collect statistics from the entire Wikipedia. Using a sample of Wikipedia, the authors reported F_1 figures of 95.8% for the task of disambiguation and 73.8% for the task of anchor identification.

Similarly to these works, we also address the problem of identifying Wikipedia anchors using a supervised approach. Since the method proposed by [Milne and Witten \(2013\)](#) achieved the best performance reported in the literature, we use it as baseline in our work.

2.6.2 Topology-based Wikification

In the topology-based wikification approach, researches assume that the input data is an article from Wikipedia already contains some outgoing links (outlinks, for short). The main purpose is to enrich existing articles with new links. It is important to observe that, in this approach, wikification is treated as a single task and disambiguation is performed implicitly. In this section, we summarize unsupervised strategies that are representatives of this approach, as reported in the studies by [Adafre and Rijke \(2005\)](#), [West, Precup and Pineau \(2009\)](#) and [Cai *et al.* \(2013\)](#).

In the early work by [Adafre and Rijke \(2005\)](#), a clustering technique was used to identify a set of articles similar to an input article based on the incoming links they share (also called inlinks). The aim is to suggest new links to an input article a (i) if they appear as outlinks in articles similar to a but not in a , and (ii) if anchor text of the similar articles is also found in a .

[West, Precup and Pineau \(2009\)](#) also considered the link content only and, in this case, used the link adjacency matrix to represent the Wikipedia graph. The authors proposed that features (outlinks) that hold for an article also hold for a similar one and, as a consequence, if most of the articles similar to a have a certain feature in common (e.g., they point to a common article), a should have that feature. In order to identify the common features, the approach projected the Wikipedia similarity matrix onto a reduced eigen-space using principal component analysis (PCA). The articles to be enriched are projected onto the same space and, then, back to the article space. This allows an error reconstruction per link to be assessed and used to rank potential new links. A qualitative evaluation showed superiority of proposed method over state of art. Besides, its predicted links were considered by evaluators as more valuable than the average links already present in Wikipedia articles.

In a more recent effort, [Cai *et al.* \(2013\)](#) proposed an iterative algorithm which extends sparsely-linked articles by adding more links. At each iteration, the algorithm uses the Wikipedia links co-occurrence matrix to provide these links. More specifically, the algorithm maintains in each iteration a concept co-occurrence matrix snapshot and uses it to disambiguate unlinked terms for the next iteration until no more links can be added. The authors reported F_1 figure of

82.58% on average when compared against state-of-art techniques.

In our work we focus on the problem of predicting which concepts should be used as anchors. Similarly to the work by [West, Precup and Pineau \(2009\)](#), we employ matrix factorization, which we used only as one prediction component of a supervised linear regression.

2.6.3 Link prediction in other domains

The problem of link prediction consists in predicting the link status of a pair of nodes of a partially observed graph — a problem tackled by recommender systems (to recommend movies, friends, co-authors, etc.) as in the work by [Koren \(2008\)](#); by social network analysis as in the contributions by [Hasan *et al.* \(2006\)](#), [Liben-Nowell and Kleinberg \(2007\)](#) and [Li, Yeung and Zhang \(2011\)](#); and by advertising click-through prediction as reported by [Menon *et al.* \(2011\)](#), to name a few. Even though many graph algorithms have been proposed to this problem, the use of latent feature models has attracted much attention as a robust and efficient way to capture patterns useful to predict the graph topology. Matrix factorization models in particular have been widely adopted in the machine learning community, specially after its successful use in the recommender system domain. As a result, several authors have proposed contributions which combine feature-based prediction, traditionally used in ML, with factor-model based prediction: examples include the studies by [Rendle \(2012\)](#) and [Menon and Elkan \(2011\)](#).

In their work, [Menon and Elkan \(2011\)](#) propose a linear factor model for the task of link prediction (classification and ranking) able to take advantage of edge and node features — they also provided a comprehensive literature review on link prediction. In a later work, [Rendle \(2012\)](#) propose a set of algorithms able to learn factor models that incorporate edge and node features in the recommender systems domain — they also contributed with a tool called Factorization Machine Library (LibFM).

Considering the valuable properties of the models proposed by [Menon and Elkan \(2011\)](#), such as scalability and appropriateness for imbalanced supervised tasks, we extended their model for solving the wikification problem as we in detail in the next chapter and also in [Ferreira, Pimentel and Cristo \(2015\)](#). One novelty of our proposed model is the handling of directed graphs which we achieve by including a latent predictor composed of two latent components: one aspect that captured the undirected aspect of link, and other that captured the residual directional aspect of the link. Unlike previous work that have used topological aspects of the Wikipedia concept graph for wikification, which is the case in the research by [Cai *et al.* \(2013\)](#) and [West, Precup and Pineau \(2009\)](#), in our mode we also make use of node and edge features and focus on the anchor prediction problem. Our proposed model outperformed the best baseline in literature, the method by [Milne and Witten \(2013\)](#). In the next chapter we detail our model and, in the following chapters, we present a comprehensive evaluation of the model. In particular, we study the impact of different training sizes, the importance of each predictor component, the importance of each link feature, the effect of ambiguity in prediction performance, and the selection of training

samples according to their quality rates.

2.6.4 *Quality of interconnected content*

As we are also interested on the impact of quality on linking prediction learning, research on quality in reference repositories is also related to our study. While this topic covers a wide variety of subjects, we here restrict our reviewing to automatic quality assessment and studies on linking and quality.

The problem of automatically determining the quality of a piece of information has long been addressed in literature. As quality can be viewed as a multi-dimensional concept, many authors have proposed general taxonomies of quality dimensions; examples include the reports by [Wang and Strong \(1996\)](#), [Tejay, Dhillon and Chin \(2006\)](#) and [Ge and Helfert \(2007\)](#). Examples of such dimensions are coherence, completeness and correctness, to cite a few. In general, to assess each of these dimensions, statistical indicators are extracted from the sources which constitute the information whose quality has to be assessed. For instance, in the specific case of collaboratively created content, indicators can be extracted from the content of the articles (e.g., structure, style, length and content readability), information about the editors (e.g. their edit history), and the networks created by links between the documents.

If we restrict our literature review only to articles about Wikipedia, the first adoption of structural properties, length, article network topology and edit history, as indicators, appears in [Dondio *et al.* \(2006a\)](#). Topology was also explored by [Korfiatis, Poulos and Bokus \(2006\)](#), [Rassbach, Pincock and Mingus \(2007\)](#), [Kirtsis *et al.* \(2010\)](#), [Tzekou *et al.* \(2011\)](#), and [Dalip *et al.* \(2009\)](#), who added, as additional sources, text style and readability. Other authors have also proposed knowledge resources and guides for developing quality measurement models, as it is the case of the contributions by [Choi and Stvilia \(2015\)](#), [Stvilia *et al.* \(2008\)](#), [Stvilia *et al.* \(2007\)](#). Methods to assess quality in general summarize these previously proposed quality dimensions and indicators. We can roughly classify these studies according to the machine learning strategy employed: (i) the unsupervised strategy adopted by [Dondio *et al.* \(2006a\)](#), [Hu *et al.* \(2007\)](#) and (ii) the supervised strategy employed by [Rassbach, Pincock and Mingus \(2007\)](#), [Xu and Luo \(2011\)](#), [Dalip *et al.* \(2011\)](#), [Dalip *et al.* \(2013\)](#).

All these previous studies assume that the manual quality assessment of Wikipedia and its associated reviewing processes lead to reliable ground truth about content quality. This idea has been challenged sometimes in literature. For instance, missing links can remain for a long time as observed by [Sunercan and Birturk \(2010\)](#), links can be biased towards popularity and importance as identified by [Hanada, Cristo and Pimentel \(2013\)](#), and links are rarely clicked as reported by [Paranjape *et al.* \(2016\)](#). This last issue is particularly remarkable as it would be expected that appropriate links should be followed once they are created. However, [Paranjape *et al.* \(2016\)](#) have shown that, in spite of the Wikipedia guidelines indicate what should be considered an appropriate link, 66% of the links created are never clicked and approximately just

1% of the remaining reached around 100 clicks. This suggests flaws in the reviewing processes.

2.7 Final Considerations

We presented the notation and background necessary for understanding this thesis. Throughout this chapter, we made a brief overview of the basic concepts and terminology including supervised learning, dyadic and link prediction, accompanied by the necessary mathematical formalism. Besides, we show how these concepts are related to our model. We also presented the quality assessment process adopted in reference collections such as Wikipedia. We finished the chapter with a review of the related literature and discussed how our model fills the gap left by existing methods.

In the next chapter we present our proposed model. Unlike previous approaches adopted for wikification, our method: (i) takes better advantage of the graph topology by describing it using latent features, (ii) provides a comprehensive study about the importance of predictors currently used in wikification and their effectiveness on dealing with ambiguous labels, (iii) and evaluate the impact of the quality of training samples on wikification. In next chapter, we present our model, along with its learning process, dyadic and monadic features.

A LATENT FEATURE MODEL FOR LINK PREDICTION IN A CONCEPT GRAPH

In this chapter, we formulate the Wikification problem as a link prediction problem, propose a prediction model to solve it, and the corresponding learning algorithm. As the model is a combination of a latent component with a feature-based component which uses human-engineered features, we also present these features.

3.0.1 Notation

We use boldface uppercase letters, such as $\mathbf{M}$, to denote matrices, and boldface lowercase letters, such as $\mathbf{v}$, to denote vectors. The i th row of $\mathbf{M}$ is denoted as $\mathbf{M}_i$ where the element at the i th row and j th column in $\mathbf{M}$ is denoted by M_{ij} .

The i th element in $\mathbf{v}$ is denoted by v_i .

Given $\mathbf{M}$ and $\mathbf{v}$, we use $\mathbf{M}^\top$ and $\mathbf{v}^\top$ to denote matrix and vector transposes, and $\text{diag}(\mathbf{M})$ to denote matrix $\mathbf{M}$ with entries outside the main diagonal equal zero. The Frobenius norm of $\mathbf{M}$ is given by $\|\mathbf{M}\|_F^2$ and L2-norm of $\mathbf{v}$ by $\|\mathbf{v}\|_2$. We denote the inner product with symbol ‘ $\cdot$ ’. Sets are represented by uppercase letters such as $\mathcal{S}$ and its cardinality by $|\mathcal{S}|$.

3.1 The Wikification Problem

At high level, the Wikipedia can be viewed as a directed graph $\mathcal{W} = (\mathcal{A}, \mathcal{L})$ where the set of nodes $\mathcal{A}$ represents the articles and the set of edges $\mathcal{L}$ represents the links between the articles. As each article is associated with a concept (expressed by its title), we refer to $\mathcal{W}$ as a *concept graph*. Figure 3 shows a portion of a concept graph with seven articles (e.g., *Charles Darwin*, *Stephen Baxter*, and *Natural Selection*) and six links (e.g., the link from *Charles Darwin* to *Evolution*).

Figure 3 – Concept graph associated with two example articles Charles Darwin and Stephen Baxter.

Source: Elaborated by the author.

Each article refers to many other concepts, expressed by means of different words and phrases, which we call *labels*. For instance, in Figure 3 the article about *Charles Darwin* refers to labels such as “Natural Selection”, “naturalist”, “evolution”, and “theory”. When an editor writes an article, he must decide which labels should be linked to appropriate Wikipedia articles. By doing that, the editor allows the readers to better understand the current article. We call these linked labels as *anchors*. For instance, from the set of possible labels in the article about *Stephen Baxter*, the underlined labels (“British” and “Evolution”) are anchors. Also, note that a concept can be referred to by different labels (e.g., *England* is referred to by both “English” and “British”) and the same label can refer to different concepts (“evolution” refers to *Evolution* and *Evolution (Baxter novel)*).

Given these initial definitions, the Wikification problem consists in determining, from a set of labels used in an article, (a) which concepts these labels refer to and (b) which labels should be anchors. As in previous works reported in the literature, in this paper we deal with a relaxed version of this problem. In particular, we treat an article as a set of labels and treat as single label two different labels of an article associated with the same concept. This is not a serious issue since editors of encyclopedias are encouraged to avoid adding to an article i multiple links to the same article j .

3.2 Wikification Matrix Factor Model

As previously defined, the wikification problem is equivalent to a link prediction problem. As such, the training set consists of a partially observed graph represented by a set of dyads (i, j) and corresponding linked status Y_{ij} . Status Y_{ij} is assigned to 1 if a link is observed from node i to j . It is assigned to 0 if no link is observed. The goal is to predict the status of edges for unobserved dyads as in the work by [Menon and Elkan \(2010\)](#).

From this assumption, we can say that the wikification problem has as input a partially observed concept graph $\mathcal{T}$ which is described by pairs of articles $\{(i, j), Y_{ij}\} \in \mathcal{T}$ where $i, j \in \mathcal{A}, i \neq j$ and $Y_{ij} \neq ?$. Our goal is to make predictions for unobserved pairs of articles. This prediction is normally based on features related to (a) pairs of articles that indicate if they should be linked (e.g., the position of the label in article i and the similarity of i and j according to their inlinks) and (b) to each isolated article (e.g., how often the article is cited and the level it appears in the Wikipedia taxonomy).

Thus, given the articles $(\mathcal{A})$, we want to define a predictor function $\hat{Y}: \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ such that the larger is $\hat{Y}_{ij}$, the larger is the probability of i pointing to j . We can use this estimate to solve the link classification problem, that is, to determine if a pair (i, j) is a link or not.

Formally, let $\mathbf{x}_i \in \mathbb{R}^d$ be a vector of features related to i , $\mathbf{x}_j \in \mathbb{R}^d$ be a vector of features related to j and $\mathbf{z}_{ij} \in \mathbb{R}^D$ be a vector of features related to pair (i, j) . A simple predictor can be obtained by combining edge and node score functions based on such features, as shown in Equation 3.1.

$$\hat{Y}_{ij}(\Theta) = f_E(\mathbf{z}_{ij}; \mathbf{w}) + f_N(\mathbf{x}_i, \mathbf{x}_j; \mathbf{v}_i, \mathbf{v}_j) \quad (3.1)$$

where Θ corresponds to the weight vectors $\mathbf{v}_i, \mathbf{v}_j$, and $\mathbf{w}$ associated with feature vectors $\mathbf{x}_i, \mathbf{x}_j$, and $\mathbf{z}_{ij}$, respectively. In this case, the domain of $\hat{Y}$ is given by weight spaces associated with features of i, j , and pair (i, j) .

In the linear case, common choices for edge and node score functions are $f_E(\mathbf{z}_{ij}; \mathbf{w}) = \mathbf{w}^\top \mathbf{z}_{ij}$ and $f_N(\mathbf{x}_i, \mathbf{x}_j; \mathbf{v}_i, \mathbf{v}_j) = \mathbf{v}_i^\top \mathbf{x}_i + \mathbf{v}_j^\top \mathbf{x}_j$. Note that, as defined, f_N does not properly capture affinities between articles i and j that could be observed through the features $\mathbf{x}_i$ and $\mathbf{x}_j$. Thus, a better choice for this function would be a bilinear regression defined as $f_N(\mathbf{x}_i, \mathbf{x}_j; \mathbf{v}_i, \mathbf{v}_j) = \mathbf{x}_i^\top \mathbf{V} \mathbf{x}_j$, where $\mathbf{V} \in \mathbb{R}^{n \times n}$ (n is the number of articles) is a set of weights between each possible pair of articles.

Although it may capture characteristics of concept graph $\mathcal{W}$ (e.g., the similarity of i and j according to their inlinks), it does not take advantage of many latent patterns present in the graph topology not easily described using explicit features. Our hypothesis is that some links are more common between articles that share these latent features.

Let $\mathbf{W} \in \{0, 1\}^{n \times n}$ represents the adjacency matrix of a sample of $\mathcal{W}$. Thus, $W_{ij} = 1$

if edge $(i, j) \in \mathcal{L}$, being 0 otherwise. One way of take advantage of latent factors hidden on underlying graph is by adding a matrix factor component in the Equation 3.1. The idea is to represent $\mathbf{W}$ by the best factorization of the form $\mathbf{W} \approx \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ that agrees with the entire graph, where $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{\Lambda} \in \mathbb{R}^{k \times k}$. Thus, the relationship between i and j may be modelled by associating a latent features of each node taking the dot-product such that $\mathbf{W}_{ij} \approx \mathbf{U}_i \mathbf{\Lambda} \mathbf{U}_j^\top$, $\mathbf{U}_i$ and $\mathbf{U}_j \in \mathbb{R}^k$, $\mathbf{\Lambda} \in \mathbb{R}^{k \times k}$. A simple predictor for an undirected graph, based on such factor model, is given by Equation 3.2:

$$\hat{Y}_{ij}(\mathbf{\Lambda}) = \mathbf{U}_i^\top \mathbf{\Lambda} \mathbf{U}_j \quad (3.2)$$

where $\mathbf{\Lambda} \in \mathbb{R}^{k \times k}$ is an arbitrary diagonal matrix and $\mathbf{U}_i \in \mathbb{R}^k$ is the i th row of matrix $\mathbf{U}$ of k latent features associated with article i . A straightforward combination of the prediction models in Equations 3.1 and 3.2 is given by Equation 3.3:

$$\hat{Y}_{ij}(\Theta) = \mathbf{U}_i^\top \mathbf{\Lambda} \mathbf{U}_j + \mathbf{w}^\top \mathbf{z}_{ij} + \mathbf{x}_i^\top \mathbf{V} \mathbf{x}_j \quad (3.3)$$

where the set of weights Θ includes $\mathbf{\Lambda}$, $\mathbf{V}$, and $\mathbf{w}$. Note that this is the link classification model proposed by Menon and Elkan (2011).

As seen, this model uses the same matrix of latent features ($\mathbf{U}$) to capture the inlink and outlink behavior of an article. This is not an issue for undirected graphs since there is no inlink and outlinks in such cases. However, concept graphs of reference collections are directed and rarely bidirectional. For instance, almost 70% of the links in Wikipedia are not bidirectional as reported by Zlatic *et al.* (2006). Thus, to apply this model to wikification, we need to extend it to support directed graphs.

To accomplish this, we have to modify the matrix factor component $\mathbf{W}_{ij} \approx \mathbf{U}_i^\top \mathbf{\Lambda} \mathbf{U}_j$, which uses the same matrix of latent features to capture inlink and outlink behavior. A simple solution would be use *two* latent matrices, $\mathbf{P}_i \in \mathbb{R}^k$ and $\mathbf{Q}_j \in \mathbb{R}^k$, to capture the different link behaviors, as proposed by Li, Yeung and Zhang (2011). The resulting factorization should be $\mathbf{W}_{ij} \approx \mathbf{P}_i^\top \mathbf{\Lambda} \mathbf{Q}_j$. This solution, however, does not capture common inlink and outlink patterns existing in articles since $\mathbf{P}$ and $\mathbf{Q}$ are now completely unrelated. To solve this issue we adopt a factor model composed of two components, $\mathbf{W}_{ij} \approx \mathbf{U}_i^\top \mathbf{\Lambda} \mathbf{U}_j + \mathbf{P}_i^\top \mathbf{\Gamma} \mathbf{Q}_j$, where $\mathbf{\Gamma} \in \mathbb{R}^{k \times k}$ is also a diagonal matrix, as $\mathbf{\Lambda}$. In this model, the first component fits the *undirected* aspect of the link between articles i and j while the second component captures the residual *directed* aspect. By adopting such solution, Equation 3.3 translates into Equation 3.4:

$$\hat{Y}_{ij}(\Theta) = \mathbf{U}_i^\top \mathbf{\Lambda} \mathbf{U}_j + \mathbf{P}_i^\top \mathbf{\Gamma} \mathbf{Q}_j + \mathbf{w}^\top \mathbf{z}_{ij} + \mathbf{x}_i^\top \mathbf{V} \mathbf{x}_j \quad (3.4)$$

To set Θ , we also include specific biases related to articles i (b_i), j (b_j) and the links between

them (b_{ij}) , which results in Equation 3.5:

$$\begin{aligned}\hat{Y}_{ij}(\Theta) = & \mathbf{U}_i^\top \mathbf{\Lambda} \mathbf{U}_j + \mathbf{P}_i^\top \mathbf{\Gamma} \mathbf{Q}_j + b_i + b_j \\ & + \mathbf{w}^\top \mathbf{z}_{ij} + b_{ij} + \mathbf{x}_i^\top \mathbf{V} \mathbf{x}_j\end{aligned}\quad (3.5)$$

To measure how good is our estimate $\hat{Y}_{ij}(\Theta)$, we use the quadratic loss function $\ell(\cdot)$. Besides, we choose a link function $L(\cdot)$ to normalize this estimate such that $Y_{ij} = 1$ if edge $(i, j) \in \mathcal{L}$, being 0 otherwise. We now can find a set of weights Θ that results in a good estimate $\hat{Y}_{ij}(\Theta)$ by minimizing $\ell(\Theta)$ with weights Θ regularized. This translates into Equation 3.6:

$$\begin{aligned}\underset{\Theta}{\text{minimize}} \ell(\Theta) = & \frac{1}{2} \sum_{ij} (L(\hat{Y}_{ij}(\Theta)) - Y_{ij})^2 \\ & + \frac{\lambda}{2} \|\mathbf{U}_i\|_2 + \frac{\lambda}{2} \|\mathbf{U}_j\|_2 \\ & + \frac{\lambda}{2} \|\mathbf{P}_i\|_2 + \frac{\lambda}{2} \|\mathbf{Q}_j\|_2 + \frac{\lambda}{2} \|\mathbf{w}\|_2 \\ & + \frac{\lambda}{2} \|b_i\|_2 + \frac{\lambda}{2} \|b_j\|_2 + \frac{\lambda}{2} \|b_{ij}\|_2 \\ & + \frac{\lambda}{2} \|\mathbf{\Lambda}\|_F^2 + \frac{\lambda}{2} \|\mathbf{\Gamma}\|_F^2 + \frac{\lambda}{2} \|\mathbf{V}\|_F^2\end{aligned}\quad (3.6)$$

where (i, j) is a pair of nodes and $\frac{\lambda}{2} \|\mathbf{U}_i\|_2$, $\frac{\lambda}{2} \|\mathbf{U}_j\|_2$, $\frac{\lambda}{2} \|\mathbf{P}_i\|_2$, $\frac{\lambda}{2} \|\mathbf{Q}_j\|_2$, $\frac{\lambda}{2} \|\mathbf{w}\|_2$, $\frac{\lambda}{2} \|b_i\|_2$, $\frac{\lambda}{2} \|b_j\|_2$, $\frac{\lambda}{2} \|b_{ij}\|_2$, $\frac{\lambda}{2} \|\mathbf{\Lambda}\|_F^2$, $\frac{\lambda}{2} \|\mathbf{\Gamma}\|_F^2$, and $\frac{\lambda}{2} \|\mathbf{V}\|_F^2$ are regularizers.

3.3 Model Learning

To solve Equation 3.6, we use a stochastic gradient descent (SGD) algorithm. SGD approximates the error of the loss function by simultaneously updating all the weights Θ of the linear estimate, according to its gradients. To determine how weights are updated we derivate the regularized loss function (Equation 3.6) with respect to Θ , obtaining the weight update expressions given in

Equation 3.7:

$$\begin{aligned}
\mathbf{U}_i &= \mathbf{U}_i - \gamma (\nabla_0 \mathbf{\Lambda} \mathbf{U}_j + \lambda \mathbf{U}_i) \\
\mathbf{U}_j &= \mathbf{U}_j - \gamma (\nabla_0 \mathbf{\Lambda}^\top \mathbf{U}_i + \lambda \mathbf{U}_j) \\
\mathbf{P}_i &= \mathbf{P}_i - \gamma (\nabla_0 \mathbf{\Gamma} \mathbf{Q}_j + \lambda \mathbf{P}_i) \\
\mathbf{Q}_j &= \mathbf{Q}_j - \gamma (\nabla_0 \mathbf{\Gamma}^\top \mathbf{P}_i + \lambda \mathbf{Q}_j) \\
\mathbf{w} &= \mathbf{w} - \gamma (\nabla_0 \mathbf{z}_{ij}^\top + \lambda \mathbf{w}) \\
\mathbf{\Lambda} &= \mathbf{\Lambda} - \gamma (\nabla_0 \text{diag}(\mathbf{U}_i \mathbf{U}_j^\top) + \lambda \mathbf{\Lambda}) \\
\mathbf{\Gamma} &= \mathbf{\Gamma} - \gamma (\nabla_0 \text{diag}(\mathbf{P}_i \mathbf{Q}_j^\top) + \lambda \mathbf{\Gamma}) \\
\mathbf{V} &= \mathbf{V} - \gamma (\nabla_0 \mathbf{x}_i \mathbf{x}_j^\top + \lambda \mathbf{V}) \\
b_i &= b_i - \gamma (\nabla_0 + \lambda b_i) \\
b_j &= b_j - \gamma (\nabla_0 + \lambda b_j) \\
b_{ij} &= b_{ij} - \gamma \nabla_0
\end{aligned} \tag{3.7}$$

where $\nabla_0 = L(\hat{Y}_{ij}(\Theta)) - Y_{ij}$, (i, j) represents a pair of articles (not necessarily linked), and γ and λ are parameters that control how large the updates are (learning factor) and how important the regularization is. Our detailed GD approach is described in Algorithm 1.

As we can note, the minimum of the regularized loss function is approximated in E steps (lines 4-29). The algorithm starts by initializing the weights in Θ with random values (line 1). At each step, a random sample of examples from the training set is picked (line 7). This means that not all examples in training set need to be computed in each step. Thus, the link prediction is computed for positive and negative example (i, j) (i.e., linked or not linked) extracted from a sample of training collection $\mathcal{T}$ (lines 8-9). According to the observed error, the gradients associated with each weight in Θ are updated (lines 10-28). As in Koren (2008), we use different parameters to control the learning rate (γ) and the impact of regularization (λ) in our implementation. We also update the learning rate along the iterations using an exponential decay strategy (line 6), as suggested in Bottou (2012).

3.3.1 Link and Article Attributes

In this section, we describe the set of attributes we have used to capture statistics of the edges and nodes from Wikipedia's concept graph. The most of them are well known attributes in wikification's literature and they were designed to model the affinity between two articles in terms of edges. Few of them attempted to characterize in terms of nodes individually. For this reason, we proposed two additional attributes to better describe the nodes: *inlink ratio* and *outlink ratio*.

Algorithm 1: Stochastic Gradient Descent algorithm to predict links in the asymmetric graph of Wikipedia

Input: Number of epochs E , Learning rates $\gamma_{\{1..3\}}$, Regularizers $\lambda_{\{1..5\}}$, % of sample $Sperc$, Link function $L: \mathbb{R} \rightarrow \{0, 1\}$, Loss function $\ell: \mathbb{R} \rightarrow \mathbb{R}$

```

1 Start  $\mathbf{U}, \mathbf{W}, \mathbf{\Lambda}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \mathbf{\Gamma}, \mathbf{b}$ , and  $\mathbf{w}$  using random values;
2 Let  $\mathcal{T}$  = set of pairs  $(i, j)$ . For each pair  $(i, j)$  where  $i$  does not link to  $j$ ,  $y_{ij} = 0$ . If  $i$  links
   to  $j$ ,  $y_{ij} = 1$ ;
3 Let  $\mathcal{S}$  a random sampling of  $Sperc$  examples from  $\mathcal{T}$ ;
4 for  $e = 1$  to  $E$  do
5   for  $n = 0$  to 4 do
6      $\gamma_n = \frac{\gamma_n}{e(1 + \lambda_n \gamma_n)}$ 
7   for each pair  $(i, j) \in \mathcal{S}$  do
8      $\hat{y}_{ij} = (\mathbf{u}_i^\top \cdot \mathbf{\Lambda} \cdot \mathbf{u}_j + \mathbf{p}_i^\top \cdot \mathbf{\Gamma} \cdot \mathbf{q}_j + b_{u_i} + b_{u_j})^\top$ 
9        $+ (\mathbf{w}_{ij}^\top \cdot \mathbf{z}_{ij} + b_{w_{ij}})$ 
10       $+ (\mathbf{x}_i^\top \cdot \mathbf{V} \cdot \mathbf{x}_j)$ 
11      $\hat{y}_{ij} = L(\hat{y}_{ij})$ 
12      $\nabla_0 = \hat{y}_{ij} - y_{ij}$ 
13      $\nabla_1 = \mathbf{\Lambda} \cdot \mathbf{U}_j$ 
14      $\nabla_2 = \mathbf{\Lambda}^\top \cdot \mathbf{U}_i$ 
15      $\nabla_3 = \mathbf{\Gamma} \cdot \mathbf{Q}_j$ 
16      $\nabla_4 = \mathbf{\Gamma}^\top \cdot \mathbf{P}_i$ 
17      $\nabla_5 = \mathbf{z}_{ij}^\top$ 
18      $\nabla_6 = \text{diag}(\mathbf{U}_i \cdot \mathbf{U}_j)^\top$ 
19      $\nabla_7 = \text{diag}(\mathbf{P}_i \cdot \mathbf{Q}_j)^\top$ 
20      $\nabla_8 = \mathbf{x}_i \mathbf{x}_j^\top$ 
21      $\mathbf{U}_i = \mathbf{U}_i - \gamma_0 (\nabla_0 \nabla_1 + \lambda_0 \mathbf{U}_i)$ 
22      $\mathbf{U}_j = \mathbf{U}_j - \gamma_0 (\nabla_0 \nabla_2 + \lambda_0 \mathbf{U}_j)$ 
23      $\mathbf{P}_i = \mathbf{P}_i - \gamma_0 (\nabla_0 \nabla_3 + \lambda_0 \mathbf{P}_i)$ 
24      $\mathbf{Q}_j = \mathbf{Q}_j - \gamma_0 (\nabla_0 \nabla_4 + \lambda_0 \mathbf{Q}_j)$ 
25      $\mathbf{w} = \mathbf{w} - \gamma_1 (\nabla_0 \nabla_5 + \lambda_1 \mathbf{w})$ 
26      $\mathbf{\Lambda} = \mathbf{\Lambda} - \gamma_2 (\nabla_0 \nabla_6 + \lambda_2 \mathbf{\Lambda})$ 
27      $\mathbf{\Gamma} = \mathbf{\Gamma} - \gamma_2 (\nabla_0 \nabla_7 + \lambda_3 \mathbf{\Gamma})$ 
28      $\mathbf{V} = \mathbf{V} - \gamma_2 (\nabla_0 \nabla_8 + \lambda_4 \mathbf{V})$ 
29      $\mathbf{b}_i = \mathbf{b}_i - \gamma_3 (\nabla_0 + \lambda_5 \mathbf{b}_i)$ 
30      $\mathbf{b}_j = \mathbf{b}_j - \gamma_3 (\nabla_0 + \lambda_5 \mathbf{b}_j)$ 
31 return  $\Theta = \mathbf{U}, \mathbf{W}, \mathbf{\Lambda}, \mathbf{V}, \mathbf{P}, \mathbf{Q}, \mathbf{\Gamma}, \mathbf{b}, \mathbf{w}$ 

```

3.3.1.1 Link attributes

These attributes are related to concept associations between articles i and j , where i is the source article and j is a candidate to destination article:

- *Link probability* $p(u)$: defined as the number of Wikipedia articles that use a label u as anchor, divided by the number of articles that mention u . Note that [Mihalcea and Csomai \(2007\)](#) estimate the link probability as the probability of a label to be an anchor.

As consequence, two attributes are used (average and maximum link probability) since different labels may have been used to mention the same concept.

- *Relatedness*: estimates how much two concepts are related, based on how many inlinks they share, by means of two attributes, $R1$ and $R2$. Given two concepts c_i and c_j , let $\mathcal{I}$ and $\mathcal{J}$ be the sets of all articles that link to c_i and c_j respectively, and $\mathcal{W}$ be the set of all Wikipedia articles. We now define the inlink relatedness between c_i and c_j as:

$$r_{ij} = \frac{\log(\max(|\mathcal{I}|, |\mathcal{J}|) - \log(|\mathcal{I} \cap \mathcal{J}|))}{\log(|\mathcal{W}|) - \log(\min(|\mathcal{I}|, |\mathcal{J}|))} \quad (3.8)$$

The relatedness, $R1_{ij}$, of a candidate concept c_j (referred by a label w in article i) is taken as the weighted average of r_{ju} to each context concept c_u , where c_u is a concept mentioned by an unambiguous anchor in article c_i . The weight w_u associated with each context concept c_u is given by $w_u = \frac{1}{2}(p(u) + \frac{1}{n} \sum_{k=1, k \neq u}^n r_{uk})$ where $p(u)$ is c_u link probability and $\frac{1}{n} \sum_{k=1, k \neq u}^n r_{uk}$ is the average relatedness of c_u to all other context concepts c_k . Finally, $R1_{ij}$ is calculated as the weighted average of a candidate sense to the context articles:

$$R1_{ij} = \frac{\sum_{u=1}^n w_u r_{ju}}{\sum_{u=1}^n w_u} \quad (3.9)$$

$R1_{ij}$ gives more importance to labels more often used as anchors and more strongly related to the central thread of article c_i . The intuition behind $R1_{ij}$ is that the more strongly a set of concepts c_u is related to the context of article c_i , the more likely c_i links to c_j if c_j is also related to c_u (according to the inlink relatedness r_{ju}). $R2_{ij}$ is a relaxed version of $R1_{ij}$ where u refers to any candidate concept in article c_i , instead of only anchors. Thus, $R2_{ij}$ is the weighted average relatedness of all concepts mentioned in c_i as observed by [Milne and Witten \(2008\)](#).

- *Frequency*: the number of times the concept c_j is mentioned in document c_i . Thus, this metric counts the frequency of link (i, j) to capture its importance and, by extension, its link-worthiness.
- *Location & spread*: set of features based on the locations where concepts are mentioned, normalized by the length of the document. These attributes are (a) *first occurrence*, since concepts mentioned in introduction tend to be more important; (b) *last occurrence*, since concepts mentioned at conclusion may be important; and (c) *spread* which measures the distance between first and last occurrences, since important concepts tend to be discussed in introductions, conclusions and consistently throughout documents, according to [Milne and Witten \(2008\)](#).
- *Disambiguation confidence*: estimate provided by the disambiguation classifier by [Milne and Witten \(2008\)](#). This classifier is trained using pairs of labels and associated concepts

observed on a sample of Wikipedia articles. Each pair is represented basically by the frequency which the label is used to represent the concept and by the relatedness of the concept. As observed for link probability, we use as attributes the average and maximum disambiguation confidence since many different labels may have been used to mention the same concept. Note we use the disambiguation confidence differently from [Milne and Witten \(2008\)](#). We here use the estimate as an attribute for each possible concept related to a label, where [Milne and Witten \(2008\)](#) kept only the concept with highest confidence value. As consequence, we have more concepts likely to be valid labels.

3.3.1.2 Article attributes

These attributes are related to characteristics of one particular concept:

- *Generality*: measures the length of the path from the root of the category hierarchy to the concept j . This allows the classifier to distinguish specialized concepts that the reader may not know about from general ones that do not require explanation, as argued by [Milne and Witten \(2008\)](#).
- *Inlink ratio*: measures the popularity of given concept j by the number of inlinks normalized by the total of links in collection.
- *Outlink ratio*: measures the number of outlinks pointed by j normalized by the total of links in collection.

3.4 Link Prediction System Architecture

Given the model and the learning algorithms for wikification, we now present a conceptual architecture for a system able to predict links for articles. Such architecture is outlined in Figure 4, which illustrates the scenario where all the links of a recently created article have to be suggested. Note, however, that other application scenarios are possible, such as to find missing links in the concept graph, to verify links that have been previously created, and to complete links existing in a recent article for which some links have already been created.

As illustrated in Figure 4, the system is composed of three main modules, (i) the Feature Extractor module, (ii) the Model Learner module and (iii) the Predictor module. Given the conceptual graph corresponding to the input dataset (illustrated on the top-left of the figure), a part of the graph is used as source of training examples, the *training sub-graph* (illustrated as darker nodes in the graph). The processing flow in the figure is as follows. From the training examples (1), the Feature Extractor module estimates the features (2) that will be used by the Dyadic Component (DC) and the Monadic component (MC) of the prediction model. The adjacency matrix of the training sub-graph is also provided as input (3) for the Latent Component (LC) of the model. Using such inputs, the Model Learner module derives the model to used by the Predictor module (6). All

Figure 4 – Conceptual Architecture for our system of Link Prediction

Source: Elaborated by the author.

these tasks can be carried out offline. When a prediction is demanded (4), given a new article (illustrated by the node in dashed line), the Predictor module uses the features of the new node (5) and the learnt model (6) to determine which concepts, candidates to be anchor text, should be really linked (7). Although in the figure it is assumed that the candidates to be anchors are known *a priori*, a real system has to adopt a heuristic to determine the candidates. A simple one would be to select all the labels in the text that are concepts in the concept graph.

3.5 Final Considerations

In this chapter, we presented a novel model to be applied to Wikification along with the corresponding learning algorithm and human-engineered features, dyadic and monadic. The model was derived from the undirected formulation proposed by [Menon and Elkan \(2011\)](#). We extended their work to support a directed graph by means of two latent factorizations which capture directed and undirected aspects of the linkage. We also presented the human-engineered features used to represent the links. To this, we adopted dyadic features commonly used in literature as well as some new monadic features. To minimize the overall prediction error, we used a gradient descent strategy. We finished the chapter outlining the architecture of a system designed to predict links based on the model. In next chapters, we present the evaluation of our model.

In the next chapter we detail the methodology and the evaluation we employed.

METHODOLOGY

In this chapter, we present the evaluation methodology we used in our experiments. In particular, we describe the test dataset, evaluation metrics, implementation details, and experimental procedures we have used.

4.1 Wikipedia School Dataset

Our collection is based on a sample of articles from the Wikipedia snapshot of December 8, 2014.¹ This sample consists of 6,000 articles commonly used in wikification research. It is referred to as “2013 Wikipedia Selection for Schools” (for short, Wikipedia School or School).² Articles which belong to this dataset were extracted from the original Wikipedia site. The selection criteria adopted by its authors took into account the importance and the quality of the articles according to the Wikipedia community – this is because the Wikipedia School is intended to be used in schools and in educational projects.

Although Wikipedia School is composed of articles which have topics of high importance and quality for educational needs, some of them were not rated for quality by the Wikipedia community. This was due to the facts that (i) most of the articles from Wikipedia lack quality ratings, and (ii) many articles in the Wikipedia School selection were classified as page lists in Wikipedia and, in Wikipedia, rates are not assigned to articles classified as page lists.

To ensure that our dataset is composed only of quality-rated articles, we removed all pages not evaluated by the community or classified as page lists in Wikipedia. After this filtering was carried out, our dataset contained 5,132 articles. We also kept all links pointing to/from articles in the sample. Finally, we removed the links pointing to articles not present in the sample.

Table 2 shows the quality rate distribution in our dataset (Wikipedia School) and in the

¹ <http://en.wikipedia.org/wiki/Wikipedia:Database_download>

² <<http://www.sos-schools.org/wikipedia-for-schools>>

snapshot of Wikipedia (English edition) from where Wikipedia School was extracted. We note that the rating distributions are very different, with School being a dataset composed of few low-quality articles (only 13.2% of ST and SB articles in School against 92% in Wikipedia). The distribution is also much more skewed in Wikipedia than in School.

Table 2 – Quality Rating Distribution for Wikipedia School and Wikipedia, English Edition

Quality ratings	Wikipedia English Ed. #articles	Wikipedia English Ed. %	Wikipedia School #articles	Wikipedia School %
FA	5,637	0.1	545	10.6
AC	1,551	0.0	29	0.6
GA	23,395	0.5	535	10.4
BC	105,790	2.3	1,945	37.9
CC	215,709	4.8	1,397	27.2
ST	1,359,114	30.1	644	12.5
SB	2,797,663	62.0	37	0.7
Totals	4,508,859		5,132	

Table 3 presents statistics associated with our concept graph. It is composed of 5,132 nodes (School articles) and about 169 thousand links (column $(i, j)^+$). If we consider all possible pairs that could be linked (an concept/article a_1 could be linked to any other concept/article a_2 if a_2 appears as a label in a_1), the concept graph could have more than 26 million links (column $(i, j)^-$). Thus, if treated as a classification problem, this is a very skewed one, with much more negative examples (1 positive example for 155 negative ones). As expected for a reference collection, School is densely interconnected, with an average of 33 links per article.

Table 3 – Topology statistics of concept graph extracted from Wikipedia School

statistic	value	description
Nodes	5,132	number of nodes in the collection
$(i, j)^+$	169,306	number of links in the collection
$(i, j)^-$	26,168,118	number of possible links considering all pairs existing in the collection
+:- ratio	1:155	ratio positive:negative examples
Average degree	33.0	average number or links per article

4.2 Evaluation Metrics

To assess the performance of the methods, we use the following metrics: precision, recall, F1, and AUC (*Area under the the Receiver Operating Characteristics Curve*). Precision, recall, and F1 are metrics derived from the four possible outcomes of a contingency matrix (links classified as links, links classified as non-links, non-links classified as links, and non-links classified as non-links), as illustrated in Table 4.

Table 4 – Contingency matrix for link classification

		Actual Class	
		Link	Not link
Predicted as	Link	True Positive (TP)	False Positive (FP)
	Not Link	False Negative (FN)	True Negative (TN)

Precision is defined as the fraction of pairs correctly assigned as link and recall is the fraction of pairs of the link class correctly classified. Given the contingency matrix, precision p and recall r are defined as:

$$p = \frac{TP}{TP + FP} \quad (4.1)$$

$$r = \frac{TP}{TP + FN} \quad (4.2)$$

To obtain a single-number summary of precision and recall, we use the F1 score, that is, the harmonic mean of precision p and recall r , given by:

$$F_1 = 2 \frac{pr}{p + r} \quad (4.3)$$

Another summarizing metric commonly used in wikification literature is *accuracy*, given by $(TP + TN)/(TP + TN + FP + FN)$. Although these metrics are widely used to assess performance in classification, they are much criticized as they are sensitive to biases (such as class distribution skewness) and do not take into account the chance level performance (POWERS, 2011). Because of this, we also report performance using AUC. Intuitively, AUC gives larger scores for methods that rank positive cases (links, in our scenario) above negative cases (non-links). It is particularly useful in situations such as wikification where the class distribution is very skewed (most of the article pairs will not be linked) as it is insensitive to imbalanced classes. We calculated AUC as defined in Ling, Huang and Zhang (2003):

$$AUC = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 n_1} \quad (4.4)$$

where n_0 and n_1 are the number of positive and negative examples, respectively, and $S_0 = \sum r_i$ is the rank of i_{th} positive example in the ranked list.

To illustrate how the previously described metrics (accuracy, AUC, and F_1) are related to each other, let us consider six classifiers, M1 to M6. They output probability estimates for 10 pairs of concepts, being 7 of them not linked while the remaining 3 are linked. Suppose that all methods classify 3 pairs as links (+) and the remaining as non-links (-). If we sort the pairs according to increasing probability of being links, we get the ranked lists shown in Table 5. From

this table, we note that while metrics Accuracy and F_1 agree in general, different rank evaluations are observed for Accuracy and AUC.

Table 5 – Accuracy, AUC, and F_1 figures obtained for classifiers M1, M2, M3, M4, M5, and M6

Method	-	-	-	-	-	-	-	+	+	+	Accuracy	AUC	F_1
M1	-	-	-	-	-	-	+	-	+	+	0.80	0.95	0.67
M2	-	-	-	-	-	-	+	+	+	-	0.80	0.86	0.67
M3	-	-	-	-	-	+	+	+	-	-	0.60	0.71	0.33
M4	+	+	-	-	-	-	-	+	-	-	0.60	0.24	0.33
M5	-	-	-	-	-	+	+	-	-	+	0.60	0.81	0.33
M6	+	-	-	-	-	-	-	+	+	-	0.80	0.57	0.67

By comparing M1 and M2 performance, we note that AUC is a intuitively better measure than accuracy. M1 and M2 are equivalent according to accuracy, as both correctly classified 80% of the instances. However, M1 clearly yields a better overall estimate than M2, since less non-links are considered more likely to be links, i.e., links are generally ranked higher than non-links in M1. The advantage of M1 is captured by AUC: 0.95 for M1 versus 0.86 for M2.

Similarly, analysing M3 and M4 AUC figures, it is clear that M3 is better than M4 since, according to M4, most of the non-links are more likely to be links than the actual links. Again, this is captured by AUC (0.71 for M3 versus 0.24 for M4). However, according to accuracy, both methods performed equally well mainly because they correctly classified the non-links, which correspond to most of the instances. M3 and M4 illustrate how AUC, different from accuracy, is not affected by an imbalanced distribution.

Finally, M5 and M6 represent a counter example, where AUC judges as best the method that achieved the higher error rate. In particular, AUC was greater for M5 than M6 (0.81 versus 0.57), although M5 presented a worse performance than M6 in terms of accuracy (60% for M5 versus 80% for M6). Despite this scenario being possible, it is unusual as show by [Ling, Huang and Zhang \(2003\)](#). In fact they formally demonstrated that, compared to accuracy, AUC is *more discriminating* and *statistically consistent*. As consequence, in this thesis, we provide performance figures mainly using AUC and F_1 .

We also report correlation between variables, using the Kendall τ Coefficient proposed by [Kendall \(1938\)](#). Kendall τ is a statistical measure used to evaluate ranking correlation, i.e., to which extent two lists are similar. Kendall τ $C(R_x, R_y)$ assumes that two rankings have the following properties: (i) $-1 \leq C(R_x, R_y) \leq 1$, (ii) $C(R_x, R_y) = 1$ if they are perfect concordant, i.e the rankings are equal and (iii) $C(R_x, R_y) = -1$ if they are perfect discordant, i.e the rankings are opposite to each other.

The terms concordant or discordant are related to the difference of positions of two observations. Let $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ be the set of observations from R_x and R_y , respectively. Any pair of observations (x_i, y_i) and (x_j, y_j) is considered concordant if rows of both elements

are concordant, that is, if both $x_i > x_j$ and $y_i > y_j$ or both $x_i < x_j$ and $y_i < y_j$. Otherwise, if $x_i > x_j$ and $y_i < y_j$ or $x_i < x_j$ and $y_i > y_j$. If $x_i = x_j$ or $y_i = y_j$, the pairs is not concordant neither discordant and the $C(R_x, R_y) = 0$. The Kendall τ Coefficient is defined as

$$\frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \quad (4.5)$$

where $n_0 = \frac{n(n-1)}{2}$, $n_1 = \sum_i \frac{t_i(t_i-1)}{2}$, $n_2 = \sum_j \frac{u_j(u_j-1)}{2}$, n_c is the number of concordant pairs, n_d is the number of discordant pairs, t_i is the number of tied values in the i -th group of ties for R_x , and u_j is the number of tied values in the j -th group of ties for R_y . We use the usual convention, such that, Kendall τ values between -0.3 and 0.3 indicate weak correlation; correlation is moderate for values ranging from -0.3 to -0.7 or 0.3 to 0.7; it is strong for values ranging from -0.7 to -1.0 or to 0.7 to 1.0.

4.3 Implementation details

The conceptual architecture presented in Figure 4 (page 52) gives a high-level description of the main modules of our link prediction system. Following that description, we implemented the prototype system used in all experiments discussed hereinafter. In this section we provide details our implementation.

4.3.1 Filtering articles

As illustrated in Figure 4, the Feature Extractor demands as input a sample of the (Wikipedia) concept graph. To build this graph, we have to access the talk pages of each article, available in the Wikipedia dump files.³ The talk pages contain, among other information, the quality rates assigned by human editors, which used on our evaluation. The dump-files *enwiki-YYYYMMDD-pages-articles.xml* and *enwiki-YYYYMMDD-pages-meta-current.xml* provide the article pages and the revision pages, respectively. Besides all encyclopedia articles, the dumps include lists, disambiguation and redirect pages. Also, the file *pages-meta-current* contains the only current revisions of all pages, including talk pages.

As a sample of Wikipedia, we adopted the articles in Wikipedia School instead of carrying out a random sampling. We did so because in a completely random sub-graph important pairs of articles, which would constitute perfect examples of links, could not be chosen. For example, it would be possible to extract “Natural Selection” without “Charles Darwin”. This could misguide the learning process by inducing the predictor to take as *non-linked* pairs of concepts that were, in fact, linked in the original collection. Among many strategies to alleviate this problem, a simple one is to choose, as seeds, the articles of a previously collected sample, which has already been built as a coherent and complete sub-graph. This is exactly the case of

³ Wikipedia dump files are available from <https://dumps.wikimedia.org>

the Wikipedia School collection with respect to Wikipedia. Thus, we extracted the list of pages of School, using a HTML parser, and stored as a *comma-separated-value* (CVS) file.

Using this list of articles, we processed the Wikipedia XML dump to extract the Wikipedia School sample with the corresponding quality rates. As result, we obtained two files: (i) a *new pages-articles* dump with only articles from the Wikipedia School, and (ii) a CSV file with article titles and its quality rates. Figure 5 illustrates the classes *Extractor of quality rates from revisions*, *Link filtering*, and *Redirect filtering*, which were implemented using the *mw.dump.processor* Python API⁴. We employed the API *mw.dump.processor* because it provides the means to build an efficient streaming XML parser able to process MediaWiki's XML dumps. This allows one to quickly process the XML dumps without dealing with streaming XML.

Figure 5 – The Filtering articles module carry out the pre-processing of the original Wikipedia XML file.

Source: Elaborated by the author.

As illustrated in Figure 5, the input for the filtering module comprises (i) the seed, i.e, the CSV file with the articles from School, (ii) the XML dump with pages of Wikipedia 2014 English Edition, and (iii) the XML dump with the article revisions. The parser class *Extractor of quality rates from revisions* is the first one to process the dumps. It parses the file *pages-meta-current.xml* (which contains the revisions and include talk pages), and extracts the quality rates of the seed pages. After that, the class *Redirect filtering* parses the dump to extract pages of redirection associated with the seed articles. Then, the class *Link filtering* removes all HTML links which

⁴ <https://pythonhosted.org/mediawiki-utilities/core/xml_dump.html>

point to articles which are not included in Wikipedia School. We observe that the input for the the parsers consists of the articles from the Wikipedia School augmented with new pages identified by the *Redirect filtering* class (i.e., the list of redirect pages). The result of this series of parse applications is the Wikipedia School sample we used in the experiments detailed in the next chapter. We also observe that in the overall process, in particular when carrying out the removal of the links and the processing of redirects, we keep the XML original format provided by Wikipedia. Moreover, the remaining pages of the pages-articles.xml file, such as lists and disambiguation pages, were maintained in new pages-article XML file.

4.3.2 Obtaining statistics from dump

In order to extract the attributes we use in our model (as detailed in Chapter 3), we used Wikipedia Miner, a toolkit freely available in the web⁵. The toolkit provides several resources for the processing of Wikipedia data such as tools for pre-processing, indexing, searching, and several algorithms to automate the tasks of disambiguation and anchor detection. Indeed, the results reported by Milne and Witten (2008) were implemented with the algorithms provided by them in this toolkit. The toolkit also offers a range of extractors for Wikipedia, which allow extracting the corresponding concept graph, taxonomy of topics of the articles, vocabulary of labels, and page types.

The extraction of the features used by the anchor detection algorithm provided in the toolkit demanded the downloading, the preprocessing and the hosting of a Wikipedia edition. To this end, we installed a Hadoop⁶ pseudo cluster, that is, a cluster simulated in a single multi-core machine. In particular, using a 3.47Ghz 6-core i7 machine with 16GB of RAM, we were able to perform all the necessary extraction tasks in the 14 GB dump corresponding to School dataset — the processing took about 5 hours.

The Wikipedia Miner toolkit performs sequentially 8 steps (which can be configured as separate Hadoop jobs) to carry out, in a scalable and timely fashion, the extraction of Wikipedia information as follows (as ordered steps): (1) page, (2) redirect, (3) labelSense, (4) pageLabel, (5) labelOccurrence, (6) pageLink, (7) categoryParent, (8) articleParent. During the overall process, it is possible to check both the current status of extraction and corresponding results. Once the extraction is concluded, a set of CSV file is available. The content of each file corresponds to the name of each step performed during the extraction. Figure 6 illustrates the extraction process and the classes used in extraction of statistics of the Wikipedia dump. Among the files available, we highlight the following:

- *articleParents.csv*: associates the article id with the ids of the categories the article belongs to;
- *categoryParents.csv*: associates the category id with the ids of the categories it belongs to;

⁵ <<https://github.com/dnmlne/wikipediaminer>>

⁶ <<http://hadoop.apache.org/>>

- *label.csv*: associates the word or phrases with statistics of use and the different sense articles it could refer to;
- *page.csv*: associates the id of each page with details like title, page type, etc.;
- *pageLabel.csv*: associates the id of each page with the list of labels that are used (in other pages) refers to the page;
- *pageLinkIn.csv*: associates id of each page with the list of pages that link to it, and to indexes of sentences where those links are found;
- *pageLinkOut.csv*: associates the id of each page with the list of pages that the page links to, and to indexes of sentences where those links are found;
- *redirectSourcesByTarget.csv*: associates the id of each article id with the ids of the redirects that target at the article;
- *redirectTargetsBySource.csv*: associates the redirect id with the id of the article it targets at.

Figure 6 – WikipediaMiner API was used to extract statistics which summarizes the structure of School collection.

Source: Elaborated by the author.

4.3.3 Extracting features from the concept graph

Our model demands the extraction of the monadic and dyadic features for whole concept graph given as input, as illustrated in the conceptual architecture depicted in Figure 4 (page 52). In Figure 7 we highlight the Features extraction and the Model Learning modules in order to illustrate that Feature Extraction module must compute the features for all pairs of concepts in the graph so as to provide the monadic and dyadic features to the Model Learning module. To achieve this, the Feature Extractor module inherits a set of methods from the *TopicDetector* class provided by the Wikipedia Miner toolkit: the class' methods provide implementation of algorithms to compute features for each concept mentioned in the article and to perform queries for statistics from the collection — in our processing we store the results both in the form of CSV summary files and as a vocabulary of labels (with the corresponding statistics) (as illustrated in Figure 7).

Figure 7 – Description of main characteristics of feature concept extraction.

Source: Elaborated by the author.

We also make use of the fact that the original TopicDetector class employs a trained classifier called *Disambiguator* to help generating estimates for the disambiguation of concepts that are used as feature. We note that the Disambiguator used by TopicDetector originally associates the concept with the highest estimate with the label (among other candidates as concepts related to this label). In our study, we extended the TopicDetector class so as to consider the estimate as an attribute of each possible concept.

The adjacency matrix of the input graph is a sparse binary matrix and was obtained by processing the pageLinkOut.csv and pageLinkIn.csv files. The matrix itself was used as topological attribute and needed no further processing for use by the model learning.

4.4 Evaluation Setup

To properly assess how the methods generalize in independent datasets, we estimate AUC and F1 values using *5-fold cross validation* (WITTEN; FRANK, 2011). In this procedure we partition the original collection into 5 subsamples. From the 5 subsamples, a single one is used as test set while the remaining 4 are used as the training set. The process is repeated 5 times, with each of the 5 subsamples used once as the test set. The partitions used for training and test were the same in all experiments for each method. Let $\mathcal{T}_i$ be the test set corresponding to the i -th cross-validation run. We obtain the AUC and F1 metrics for $\mathcal{T}_i$ as the average of all AUC and F1 metrics obtained for every article in $\mathcal{T}_i$. Finally, the reported AUC and F1 for the entire cross-validation is taken as the average of the AUC and F1 values obtained for $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4$,

and $\mathcal{T}_5$.

To ensure that the differences among methods we compare are statistically significant, we use the standard error (CAMPBELL; SWINSCOW, 2011) with confidence level of 95%. The standard error figures were calculated over articles F1 and AUC to ensure normally distributed values.

We also note that, since SGD does not need to be trained with large amounts of data to reach a good performance, in some experiments we did not use the entire training set available in the cross-validation. If no additional information is provided, the reader should assume that the reported SGD results were based on a random sample composed of about 50% of the training instances. We use as training data the pairs of articles that could be linked. To select such pairs, for each article a_i , we pick up the training pairs (a_i, a_j) , where a_j appears at least once as a label in a_i . In this way, we avoid placing links to a target article whose concept does not appear as a label in the source article.

Before performing the experiments, we standardized and scaled the numeric features to avoid their values varying widely. The idea is transforming the values so that they maintain their general distribution and ratios. In particular, we apply a Z-score normalization, that is, we subtract all feature vectors from their means and divide the resulting values by their standard deviations. This way, all features are centered around zero and have unit-variance. The scaling parameters are derived from the training data.

In all experiments, we use the best selection of SGD parameters (learning rate λ and regularizer factors γ) learned in separated validation sets. In cross validation, we use different learning rates and regularizers for side-information and latent weights. For each turn, a different initialization of the weight matrices was used. In all experiments, the number of latent dimensions k was set to 5. This value was derived from previous studies and constituted a good trade-off between space and performance.

The experiments were run in the same machine we use to pre-process and extract statistics from XML dump files — a 3.47Ghz 6-core i7 machine with 16GB of RAM. Our model was totally implemented in MATLAB 2013a⁷. Our main baseline is the work by Milne and Witten (2013) and, in our experiments, we used their publicly available code.⁸ In particular, as we are interested on the task of anchor prediction, we selected the anchor detector from the toolkit. Henceforth, we denote the anchor detector baseline as MW2013. We also use other algorithms available in the toolkit, since they are necessary to the anchor detector and to the feature extractor as, for instance, the disambiguator of anchor candidates and the estimator of relatedness between concepts.

⁷ <<https://github.com/raoniferreira/wikipedia-linkpred>>

⁸ <<https://github.com/dnmilne/wikipediaminer/wiki/Downloads>>

4.5 Final Considerations

In this chapter, we described the dataset used in our experiments as well as our empirical evaluation methodology, including implementation details. Among the observations made in this chapter, we highlight: (i) while our collection represents a consistent sub-sample of Wikipedia, it is biased towards high quality documents; (ii) although accuracy is a metric usually adopted in previous work, it is not the most appropriate performance measure for wikification. Thus, we report our results mainly using AUC and F_1 ; (iii) we provide some details we took in account during the implementation of prototype system — APIs used and tools which was implemented; and (iv) we adopted standard procedures for ensuring generalization (cross validation), assessing statistical significance of results (standard error), normalizing data (Z -score), and tuning parameters (exhaustive searching on validation sets). In next chapter, we present and discuss the results of our experiments.

In the next chapter we detail and discuss the results we obtained from our evaluation efforts.

EXPERIMENTS AND RESULTS

In this chapter, we present and discuss our results. We start by comparing our approach with previous state-of-art methods in the task of anchor prediction. After that, we evaluate the utility of each predictor component of our model. For the components based on features, we also study the impact of each class of features. We then investigate how our model deals with ambiguity, a common issue on wikification. Finally, we analyse the impact of the quality of the training examples in anchor prediction.

5.1 Comparison with previous models

To assess the performance of our proposed model, we start by comparing it with two baselines. The first one, which we refer to as *feature model*, is a model purely based on features which has been used in related tasks such as collaborative filtering and link prediction in social network (CHU; PARK, 2009; YANG *et al.*, 2011). In this approach, the variable to be predicted is seen as a linear combination of attributes associated with nodes (in a bilinear setting) and links (pair of nodes). As the loss function used is the logistic, this method is called logistic regression with a bilinear component. This is a very common approach for general-purpose prediction and corresponds to the model described by Equation 3.1. Our second baseline is the method proposed by Milne and Witten (2013) that, as far as we know, is the one with best performance reported in literature for the task of wikification. We refer to this method as *MW2013*. MW2013 learns a decision tree based on the features associated with the nodes and with the links. A path from the root to a leaf on the learned tree constitutes a classification rule able to distinguish pairs of nodes as link or not link. For the learner, we used the implementation of algorithm C4.5 provided by Milne and Witten (2013).

In Figures 8 and 9 we compare these two methods with our approach. We refer to our model as *latent+feature*. It corresponds to the predictor described by Equation 3.5. More specifically, the figure shows AUC and F_1 average values obtained for the three methods on

the test dataset. In this experiment, the methods were trained using from 10% to 100% of the training samples. This way, besides comparing the methods, we can also inspect their predictive performance as more training samples are available. The figure also provides standard error values considering a 95% confidence level.

Figure 8 – The performance achieved in the test set when used fractions of training set: AUC

Source: Elaborated by the author.

Figure 9 – The performance achieved in the test set when used fractions of training set: F_1

Source: Elaborated by the author.

As we can see in the figures, our latent+feature model outperforms the baselines when trained with more than 20% of the training samples, both in AUC and F_1 . It is also important to observe that our method, using only 30% of the training samples, was able to outperform the best results of both baselines, i.e., the results obtained by them when trained using all the available training samples. Another interesting characteristic of latent+feature model is its ability to continuously learn useful patterns as the training set increases. As consequence, gains over the baselines are

increasingly larger for increasingly larger training dataset. For instance, our latent+feature model trained using all available training samples reached gains about 2% and 13% in AUC and F_1 , respectively, over best reported method in wikification literature — the MW2013 model. This is not the case, for instance, of the logistic regressor (i.e feature model) which quickly reached its peak performance, not being able to take advantage of additional training examples.

When comparing the AUC and F_1 metrics, we observe that the performance of MW2013 is lower for F_1 than for AUC. As consequence, MW2013 was competitive with logistic regression only when compared using the AUC metric. Regarding the error, we note that the smaller is the training sample size, the higher is the standard error. The comparison of our model with the feature model suggests that the latent component is able to take advantage of additional data, as both AUC and F_1 increase when more training data is available. We also note that the standard error is in general smaller for our model. To better understand the possible reasons for such improvements, in next two sections we carefully study the impact of predictor components, attributes and the effect of ambiguity.

5.2 Analysis of the prediction model components and its attributes

As previously described and illustrated in the architecture discussed in Section 4.3, our latent+feature model is composed of three prediction components: latent, monadic and dyadic. The latent component corresponds to the set of latent features derived from the entire concept graph. The monadic component is associated with the article features whereas the dyadic component is associated with the link features. We now characterize our model by analysing the relative importance of each component.

To analyse the importance of an individual component C , we have configured our model such that it is composed (i) by only C or (ii) by all components except C . This way, we can infer the importance of C when taken in isolation and when removed from the model. The result of this analysis is summarized in Table 6, where columns *Single* and *Excluded* present the average performance observed on test sets for each predictor component, taken in isolation or excluded, respectively. The line indicated by *All* presents the results obtained by the combination of all components and corresponds to the complete model described by Equation 3.5. Performance figures in the table are given in AUC with 95% confidence intervals. Although not shown, similar results were obtained using F_1 .

We first note that the best overall result was obtained by the complete model (*All*). Among the models based only on a single component, the dyadic was the best with a loss of 8.2%. The second best was the Latent based model with a loss of 17.4%. We highlight the effectiveness of the latent component. In spite of compressing the information of the (observed) concept graph in a small $k \times k$ -matrix, it was able to reach a performance only 9% worse than

Table 6 – Classifier performance (AUC with 95% confidence intervals) for models composed of a single predictor component C and all components except C , where C is *Dyadic*, *Latent* or *Monadic*. Line *All* indicates the model composed of all components.

Component	Single		Excluded	
	AUC	Gain(%)	AUC	Gain(%)
All	0.912 ± 0.003	-	0.912 ± 0.003	-
Dyadic	0.837 ± 0.044	-8.2	0.749 ± 0.005	-17.9
Latent	0.753 ± 0.004	-17.4	0.894 ± 0.005	-2.0
Monadic	0.585 ± 0.000	-35.6	0.907 ± 0.006	-0.5

the dyadic component, which stores a much larger amount of information about the links. Unlike the previous components, the monadic presented a very poor prediction performance, with a loss in AUC of 35.6%.

As in the previous analysis, dyadic and latent components were the most important when removed from the model. The removal of the dyadic component resulted in a 17.9% loss in AUC, while the exclusion of latent component led to a 2% loss. These results imply that, as expected, most of the information on the latent component is also present in the dyadic component. Also, the choice of k in the latent component led to a loss of useful information observed in the dyadic component. However, the latent component clearly improves the overall performance when combined with the dyadic component. This suggests that the latent component is able to capture patterns not observed in the dyadic component.

The result obtained for the monadic component was not statistically significant as its standard error overlaps with the baseline figure (0.912 ± 0.003 for *All* against 0.907 ± 0.006 for *Monadic*). In other words, the monadic component has little to no contribution to the complete model.

We now study the impact of the dyadic attributes. To infer the impact of the attributes, we study the prediction performance (i) after adding them to the model based only on the latent component (we discarded the monadic attribute from this analysis because it gave a poor performance in previous analysis) and (ii) after removing them from the model based on the complete model (*All* in Table 6).

Table 7 summarizes the impact of each attribute when included in the model. Results are given in AUC and F_1 . As we can see, all the dyadic attributes, taken in isolation, improved the performance of the basic latent predictor. None of them, however, was able to surpasses the combination of latent and dyadic components (cf. Table 8). Among the attributes, *Relatedness*, *Link probability*, and *Disambiguation confidence* are the ones with higher impact. Attributes *Location* and *Frequency* have little impact on the result. This suggests that the relatedness between a pair of concepts and its probability of being observed as an anchor in the past constitute better pieces of evidence to distinguish what should be a link than the location and

frequency of the pair of concepts. F_1 results basically mirror what is observed with AUC.

Table 7 – Attribute impact when added to latent based model. Confidence intervals are given for a 95% confidence level.

Attribute	Added to the model			
	AUC	Gain(%)	F_1	Gain(%)
Only Latent	0.753 ± 0.004	-	0.432 ± 0.006	-
Relatedness	0.893 ± 0.002	18.6	0.625 ± 0.007	44.7
Link probability	0.850 ± 0.002	12.9	0.591 ± 0.006	36.8
Disambiguation confidence	0.829 ± 0.003	10.1	0.540 ± 0.005	25.0
Frequency	0.799 ± 0.002	6.1	0.475 ± 0.009	10.0
Location & spread	0.786 ± 0.004	4.4	0.461 ± 0.009	6.7

Table 8 is similar to Table 7, but infers the impact of an attribute by removing it from the model. Unlike the previous table, the smaller the metric value is, the bigger the importance of the attribute. General conclusions taken from this table are very similar for AUC and F_1 . As we can see, only *Relatedness* and *Link probability* provide unique information that, if omitted, leads to performance degradation (-3.6 and -1.6 losses in AUC, respectively). The other attributes, when removed, have no significant impact on the prediction performance. Their results are all statistical ties. This suggests they do not provide information that is not already provided by other attributes.

Table 8 – Attribute impact when removed from model based on latent and dyadic features. Confidence intervals are given for a 95% confidence level.

Attribute	Removed from the model			
	AUC	Gain(%)	F_1	Gain(%)
All	0.912 ± 0.003	-	0.695 ± 0.004	-
Relatedness	0.879 ± 0.002	-3.6	0.636 ± 0.006	-8.5
Link probability	0.897 ± 0.004	-1.6	0.657 ± 0.004	-5.5
Frequency	0.912 ± 0.003	0.0	0.693 ± 0.006	-0.3
Location & spread	0.912 ± 0.003	0.0	0.694 ± 0.004	-0.1
Disambiguation confidence	0.913 ± 0.003	0.1	0.695 ± 0.002	0.0

In sum, among the attributes, *Relatedness* and *Link probability* are the best ones. This time, *Disambiguation confidence* was not so useful, which was also the case for *Location* and *Frequency*. In particular, we note that *Disambiguation confidence* performed better when combined with latent factors than when excluded from the complete model. This suggests that *Disambiguation confidence* provides complementary information to that provided by the latent factors. When removed from the complete model, its impact is not important because of its redundancy regarding other attributes as *Relatedness*. In fact, the small gains observed in Table 8 indicate that all the attributes carry dependent information. For instance, disambiguation information is

probably present in *Disambiguation confidence* and *Relatedness*. Clearly, *Relatedness* and *Link probability* are very correlated.

In the next section we further study the impact of ambiguity on our model.

5.3 Impact of ambiguity on link prediction

An important problem in wikification is the disambiguation of labels into concepts (eg., determine that the label “java” in the article about programming languages refers to “Java programming language” and not to “Java island”). In fact, most of the previous approaches in the literature disambiguate the labels before classifying them as anchors. Unlike these methods, our approach does not *explicitly* disambiguate concepts. In spite of that, it is able to achieve a good performance even when using less information. In this section, we study our approach regarding ambiguous concepts to better understand its performance.

Before analysing the performance of our model, we recall that the degree of ambiguity A_c of a concept c is given by the average of senses (concepts) related to the labels used to represent c . Thus, if every label associated with c has a single concept, the value of A_c is 1 and we say that c is not ambiguous. In this section, we are interested in the performance of the methods regarding concepts c such that $A_c > 1$.

Figure 10 summarizes the performance of our model when applied to increasingly ambiguous concepts. To this, we grouped 50,000 pairs of concepts (extracted from the pooled predictions of all test sets) into four bins which correspond to the A_c intervals $(1, 2]$, $(2, 3]$, $(3, 4]$, and $(4, \infty)$, respectively. For each interval, the figure shows the proportions of hits (true positives and true negatives, in light grey) and misses (false positives and false negatives, in dark grey).

Figure 10 – Proportions of hits and misses, obtained by our complete model, for increasingly ambiguous labels.

Source: Elaborated by the author.

As shown in the figure, our model performs equally well independently on the degree

of ambiguity of the labels. In other words, it is little affected by ambiguity. Since we did not classify ambiguous concepts before detecting anchors, our model was able to naturally learn how to deal with ambiguity. In fact, this was somewhat expected because we included in the model some attributes useful to recognize ambiguity, in particular, *Disambiguation confidence* and *Relatedness*. From now on, we refer to these two attributes as *disambiguation attributes*.

To better understand the impact of label ambiguity, we now analyse the performance of disambiguation attributes when applied to ambiguous labels. We will also observe the performance of the latent component in the same scenario. In particular, we experimented two versions of the model: (i) using only disambiguation attributes; (ii) using only the latent component. Figure 11 and Figure 12 summarize the performance of these two versions when applied to increasingly ambiguous concepts. We experimented with the same pairs used in Figure 10.

Figure 11 – Proportions of hits and misses for increasingly ambiguous labels: only disambiguation features

Source: Elaborated by the author.

As seen in Figure 11, the predictor based only on disambiguation features is not much effective to distinguish anchors as labels become increasingly ambiguous, except for the last bin. This is due to the fact that the degree of ambiguity of labels is weakly correlated to the property of labels being or not anchors. More precisely, we obtained a Kendall τ value of -0.108 for the specific sample shown in Figure 11. The unusual behaviour observed in the last bin corresponds to an exception associated with the small size of this bin as concepts with more than four senses are very rare.

Figure 12 shows the performance of the predictor based only on latent features. This is the best among the two versions of the model in this scenario. Its good performance in distinguishing anchors from not anchors results from the fact that the latent component compresses a rich information about the graph. What is surprising is its stable behaviour disregarding the degree of ambiguity of the labels, as the method seems to be little affected by ambiguity. This suggests that the latent component naturally deals with ambiguity. This is probably due to the fact that latent

Figure 12 – Proportions of hits and misses for increasingly ambiguous labels: only latent component

Source: Elaborated by the author.

features capture the topology of the concept graph even if it is represented in a reduced space. More specifically, the context information associated with a concept is naturally captured by the graph topology, as we expect that different senses of a label are “located” in different regions of the concept space. Similarly, related concepts are clustered in the same regions of the latent space.

In general, we observed that the more ambiguous the concepts are, the larger the proportion of misses is. The latent component dealt very well with ambiguous concepts. When combined with all other sources of evidence, the (complete) model was able to steadily distinguish anchors independently of their degree of ambiguity.

In the next section we further study the impact that the quality rates of the training samples have on our model.

5.4 Impact of training samples quality rates on link prediction

The Wikipedia linking guideline¹ recommends the placement of links if they are relevant and helpful in the context, since excessive linking can be distracting and slow the reader down. On the other hand, redundant links clutter the page and make future maintenance harder.

While both underlinking and overlinking should be avoided, overlinked articles make it difficult for users to identify that links likely to aid their understanding. This is a common problem in Wikipedia, according to a study of log data conducted by Paranjape *et al.* (2016). The authors found that most of the links placed by editors are never or rarely clicked. Thus adding

¹ <https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking>

more links does not increase the clicks taken from a page. On the contrary, links compete with each other for user attention.

A possible reason for the finding by [Paranjape *et al.* \(2016\)](#) is that linking guidelines are not satisfactorily followed by Wikipedia editors. This is not surprising if we note that such guidelines are really reinforced for few articles. Although anyone can rate an article for its quality, strict control is only provided for A-, GA-, and FA-class articles. In particular, A-class articles (AC) require the agreement of at least two editors. GA and FA quality classes are assigned only after a review is conducted to approve the intended rating. That review is performed by a review committee or by the editors of a WikiProject, who bear ultimate responsibility for resolving disputes.¹

In sum, only high quality articles (AC, GA, and FA) are carefully inspected on its compliance to the Wikipedia manual of style, which includes detailed linking guidelines. In spite of that, previous work in Wikification literature regard any article as an equally good source of evidence about linking. As such observation deserves some caution, we now investigate the impact on wikification of the quality of the articles selected for training.

To accomplish this, we randomly sampled 30 sets of 1200 articles clustered into six groups: (i) high-quality test dataset: 200 articles with quality ratings AC, GA, and FA; (ii) high-quality training dataset: 200 articles with quality ratings AC, GA, and FA; (iii) BC-quality training dataset: 200 articles with quality ratings BC; (iv) CC-quality training dataset: 200 articles with quality ratings CC; (v) low-quality training dataset: 200 articles with quality ratings ST and SB and (vi) random training dataset: 200 articles with random quality ratings.

Using the aforementioned training and test samples, we evaluated the performance of our complete model to predict anchors. Table 9 shows the results obtained using performance measures AUC, precision and recall. It also provides standard errors calculated for a 95% confidence level. To provide a broad view, the table includes the corresponding quality class distributions in Wikipedia English Edition dataset and the Wikipedia School dataset. The first column identifies the quality of the articles used in the training sets. In all cases, only high quality articles were used in the test sets. First line (Random) presents the traditional training scheme adopted in literature, that is, training samples are randomly selected without taking to consideration their quality rating.

Considering AUC, we first note that the model trained with the random dataset was outperformed by the one trained with the high quality dataset (FA, AC, GA). Although the difference was statistically significant, the gain was rather small to, at first glance, justify the selection of high quality training samples. However, we also note that the performance obtained using the random training is more similar to the ones achieved with BC and CC training. This was expected as BC and CC are the most common quality classes in the school dataset (about 65% of the samples). As previously mentioned, School is a dataset biased towards high-quality articles. Thus, such distribution is unlikely to be found in a random sample extracted from the

Table 9 – Performance of anchor classifier according to the quality rate of the training samples measured using AUC, precision and recall with standard error calculated considering 95% confidence levels, along with the corresponding distribution in Wikipedia English Edition and Wikipedia School datasets.

Quality of training samples	Corresp. distrib. in Wikipedia (%)	Corresp. distrib. in School (%)	AUC	Precision	Recall
Random	-	-	0.9038 ± 0.0007	0.698 ± 0.004	0.720 ± 0.007
FA, AC, GA	0.7	21.6	0.9053 ± 0.0007	0.717 ± 0.007	0.706 ± 0.010
BC	2.3	37.9	0.9032 ± 0.0007	0.703 ± 0.005	0.710 ± 0.008
CC	4.8	27.2	0.9024 ± 0.0005	0.692 ± 0.005	0.730 ± 0.007
ST, SB	92.2	13.3	0.8949 ± 0.0008	0.675 ± 0.005	0.718 ± 0.007

original Wikipedia. As consequence, a random training performed using Wikipedia articles is indeed more similar to a low quality training using School articles.

To infer how a high and a random quality training would perform in Wikipedia, we now compare our predictors trained with high and low quality School samples. By doing so, we still observe a small difference in AUC. However, the difference in precision is about 6% (at the expense of a small loss in recall). Since the precision estimates the proportion of labels that the method correctly predicts as anchors, the larger is the precision, the less likely is an overlinking. In other words, the training with high quality samples leads to more precise anchor suggestions. This way, the predictor is more appropriate to support authorship, as it can assist an editor recommending links while avoiding overlinking at the same time.

Finally, the similar results obtained with high quality training (FA, AC or GA) and medium quality training (BC or CC) suggests that either (i) the casual editor and the expert reviewer have a similar understanding about what should be a proper link, (ii) a proper linking style is not determinant on the distinction of medium and high quality classes, or (iii) linking is so subjective that it is hard to assess its quality once some basic criteria are met (for instance, no excessive redundancy, exclusion of common elements such everyday words, dates, major geographic features, etc). Overall, the apparent little use of links in Wikipedia identified by [Paranjape *et al.* \(2016\)](#) indicates that more research is necessary so that this issue can be better understood.

5.5 Final Considerations

In this chapter, we presented and discussed the results of wikification models on anchor prediction task. We have shown that our model outperformed the baseline even using only 30% of training samples. The dyadic was the best performer among the prediction components. However, it did not surpass the combination with latent component, which indicates that they carry complementary information. Among the dyadic features, *Relatedness* and *Link probability* had the largest contributions. The model was also able to perform well even when dealing with very

ambiguous concepts. Finally, the selection of training samples according to their quality was effective for the most contrasting quality rates.

In the next chapter we summarize the contributions of our work while we revisit the research questions we identified from the presentation of our hypotheses. We also point out to future directions.

CONCLUSIONS AND FUTURE WORK

In this work, we proposed a wikification prediction model that combines traditional feature prediction based on dyadic and monadic components with a latent prediction component. To accomplish this, we addressed the problem of anchor detection as a link prediction in a concept graph. Latent features were then learned from the matrix representing the graph, such that the combination of the extracted latent vectors with traditional dyadic and monadic features minimized the overall prediction error. In other words, a matrix factorization was performed such that an approximation of the article-article link matrix was obtained by minimization of the reconstruction error, using a gradient descent algorithm. To evaluate our model, we carried out experiments using a sample of the Wikipedia.

This research was motivated by some questions for which we now provide answers in the following paragraphs.

How does perform a Wikification model that combines latent- and feature-based prediction components? Do these components provide complementary information such that the combined model is more accurate than its individual components?

Our model outperformed the baselines even when trained using only 30% of the training samples. It reached gains of about 13% in F_1 and 2% in AUC. Its latent component was clearly able to take advantage of additional data, as its performance increased when more training data was available. We also noted that the standard error was in general smaller for our model.

Among the prediction components, the dyadic component performed the best, followed by the latent component. The monadic component had little to no contribution to the overall model performance. Although much redundant information is captured by the components, their combination was always able to provide the best results, which indicates that they are complementary. Regarding the dyadic attributes, the best ones were *Relatedness* and *Link probability*. *Disambiguation confidence* provides complementary information to latent factors. However, the same information seems to be provided by *Relatedness*. As observed in the

components, all attributes are very dependent among them.

As the latent features represent a reduced version of the original concept graph, do they naturally deal with ambiguity, as different concepts are “located” in different regions in the concept space?

We observed that the precision degrades as the degree of ambiguity increases. However, giving that this effect is very weak we can conclude that our model is able to steadily distinguish anchors independently of their ambiguity. Among the components, it is clear that the latent component deals very well with ambiguous concepts, even though it is not associated with an explicitly disambiguation process.

What is the impact on link prediction of the selection of training samples according to their quality, as assessed by human reviewers?

From our experiments on the selection of training samples according to their quality rates, we observed that our model trained with high quality samples outperformed the model trained with low quality samples, achieving an additional gain of 6% in precision. Thus, the training with high quality samples led to more precise anchor suggestions and less overlinking. Additionally, we observed very small differences among models trained using high quality and medium quality samples. In concordance with the apparent little use of links in Wikipedia identified by [Paranjape et al. \(2016\)](#), these results suggest that the linking guidelines (or the Wikipedia reviewing process) are not as effective as they should be. More research is necessary to better understand this issue.

6.1 Limitations of this work

During this research, we have faced some difficulties and, as a consequence, our results present some limitations. Among them, we cite:

- To make feasible this study, many of our conclusions were based on a limited set of optimization parameter values. Only experiments on performance were based on a systematic study to find optimal values to parameters such as learning rate, regularizer coefficients, and the number of latent dimensions. Later experiments (such as the ones on importance of attributes, effect of ambiguity, and selection of training samples) were based on more limited sets of parameter values.
- This study was focused on collaboratively created reference datasets characterized by encyclopedic linkage. Our conclusions are probably not valid for other datasets composed by semantic connected articles, such as regular websites and Q&A Forum sites. Even for collaborative created encyclopedias, the generality of our findings should be taken with some caution as we experimented with a sample of Wikipedia. More general conclusions require the study of other datasets.

- Our experiments on the selection of training samples, according to their quality, was restricted to datasets that employ quality assessment systems similar to the one adopted by Wikipedia. And, even for Wikipedia, the results are approximated, as we used a sample (Wikipedia School) biased towards high-quality documents.

6.2 Future work

Along this work, many new questions arose. Also, some of our conclusions were conditioned to limitations in our methods and experiments. These issues have motivated new research to be pursued in the future, summarized in this section.

6.2.1 *Evaluation of the model on different domains and datasets*

Regarding datasets, the Wikia service¹ provides many freely available reference collections which could be benefited from an automatic wikification tool. As other examples, the DBPedia and Tweeter have been used as platforms of automatic linking by [Mendes et al. \(2011\)](#) and [Maio et al. \(2016\)](#), respectively. In general, our proposed model may be also applied to any link prediction problem characterized by directed links. For instance, it is easy to imagine directed graph in social network scenarios where the link prediction could evolving along the time. Regarding this scenario, latent aspects from matrix factorization can be applied to model a set of complex criteria which explain who should follow who in the future.

6.2.2 *New features*

In this work, we have implemented the features most commonly used in literature. However, many other features have been proposed in the past, such as the use of the navigation actually carried out by users as in investigation by [West, Paranjape and Leskovec \(2015\)](#). In future, it would be interesting to study the impact of these features on the effectiveness of the prediction components, in special, the monadic.

6.2.3 *Training sample selection based on quality*

The impact of the selection of articles according to their quality should be confirmed in another sample of Wikipedia. Our conclusions were drawn from the School sample which is clearly biased towards high quality content.

This research could be also extended by the analysis of the impact of selecting samples for training using automatic quality assessment methods. There is a wide literature on this topic which could be explored as discussed in Section 2.6.4 and illustrated by the contributions by [Han, Chen and Wang \(2015\)](#), [Dalip et al. \(2014\)](#), [Dalip et al. \(2011\)](#), [Dondio, Barrett and Weber](#)

¹ <http://www.wikia.com/fandom>

(2006), Dondio *et al.* (2006b). The adoption of automatic methods would allow training datasets with size not limited to the amount of articles manually assessed by reviewers.

6.2.4 Better understanding of which should be considered appropriate linking

Links are placed in articles mainly to provide to the reader the opportunity to acquire a deeper understanding of a concept. Thus, it is expected that they should be followed once they are created. At first glance, the evidence that most of the links in Wikipedia are not used, as identified by Paranjape *et al.* (2016), suggests that these links are not appropriate. However, as far as we know, no previous work has really characterized (a) which should be a proper link in Wikipedia, from a quantitative point of view; and (b) how effective is the Wikipedia reviewing process in fixing links.

These two issues could be addressed using statistical information available about the use of links and about the review process. For instance, (i) link usage should be explored as indicative of appropriateness, after discounting biases towards popularity and importance; (ii) changes in links can be tracked using the review history, such that insertions and deletions of links can be studied.

Such research could lead to the revision of the Wikipedia linking guidelines, the adoption of new criteria to be followed by reviewers when inspecting linking styles, and the design of automatic tools for assessing the appropriateness of the observed linking style. Regarding wikification, new strategies for selecting training samples could be devised with the advantage of being specific for links and not related to the overall content.

6.2.5 Investigate the use of more than one language

Our studies have considered articles written in English. Given that the Wikipedia currently has contents in almost 300 languages, articles in more than one language could be used to confirm or to complement the identification of both source anchors and destination anchors. One approach could be to evaluate our proposal in one dataset containing articles in a different language, which would be similar to the work by Li, Sun and Datta (2013) who evaluated their framework for word-sense disambiguation in English and in Traditional Chinese. Another approach would be to use datasets with more than one language, as it was the case in the work by Lawrie *et al.* (2015) when investigating cross-language person-entity linking, and of the contribution by Tsunakawa, Araya and Kaji (2014), who managed to transfer intra-language links from English to Japanese.

6.2.6 Adoption of a bipartite ranking approach

In our model, despite using a regression approach, we treated wikification as a classification problem. As previously explained, the larger is the model estimate $\hat{Y}_{ij}$, the larger is the probability

of i pointing to j . We used this estimate to solve the link classification problem. Although efficient, this approach is somewhat naive. For instance, observe the labels “author”, “evolution”, “evolution theory”, and “Natural Selection” in Figure 3. It is clear that “author” should not be an anchor while “Natural Selection” should be. However, it is not so simple to determine, among “evolution” and “evolution theory”, which one would be taken as anchor or even if any of them should be. This gradation regarding the possibility of a label to be an anchor indicates that this problem could be reformulated as a bipartite ranking problem (i.e., a ranking problem in which instances come from two categories). As such, given all labels of an article, they could be ranked according to their probabilities to be anchors.

As a bipartite ranking problem, we define a ranking function R , such that the rank value of pair (i, j) , $R(i, j)$, is greater than $R(i, k)$, if i links to j but not to k ($i \neq j \neq k$). While learning R could lead to more reliable estimates on the existence of links, this approach requires *two links per instance* to learn the link order. Such learning approach, known as *pair-wise ranking formulation*, is more expensive than the traditional classification formulation which requires just one link. The ideal would be to use a prediction function as effective as a pair-wise formulation but able to learn from the same set of instances used in the classification formulation.

This is indeed possible as it has been demonstrated by [Ertekin and Rudin \(2011\)](#). Their idea is that, to ensure that estimates for positive examples are greater than for negative ones, we should define a loss function which penalizes low estimates (especially if negatives) for positive examples and high estimates (especially if positives) for negative examples. More formally, let $\mathcal{P}$ be a set of pairs in $\mathcal{L}$ (positive examples) and $\mathcal{N}$ be a set of pairs not in $\mathcal{L}$ (negative examples). With each example $\mathcal{P}_i$ is associated a value $y_i = +1$ while a value $y_k = -1$ is associated with each $\mathcal{N}_k$. Given two examples $\mathcal{P}_i$ and $\mathcal{N}_k$, we want that $\hat{y}_{\mathcal{P}_i} \geq \hat{y}_{\mathcal{N}_k}$. For a set of weights Θ , such a function is given by Equation 6.1.

$$\ell(\Theta) = \sum_i e^{-\hat{y}_{\mathcal{P}_i}(\Theta)} + \frac{1}{p} \sum_k e^{p \hat{y}_{\mathcal{N}_k}(\Theta)} \quad (6.1)$$

where p is a factor that controls how much negatives (in the top of the ranking) are penalized. For $p = 1$, the function to be minimized is the same of AdaBoost as shown by [Ertekin and Rudin \(2011\)](#). In practice, high values of p lead to increasing large separations between negative and positives, as negatives in the top of the ranking are more rigorously penalized. By incorporating

Equation 6.1 into our model, given by Equation 3.6, we obtain:

$$\begin{aligned}
 \underset{\Theta}{\text{minimize}} \ell(\Theta) = & \sum_{ij} \frac{1}{\phi_{ij}} e^{-\phi_{ij} Y_{ij} L(\hat{Y}_{ij}(\Theta))} \\
 & + \frac{\lambda}{2} \|\mathbf{U}_i\|_2 + \frac{\lambda}{2} \|\mathbf{U}_j\|_2 \\
 & + \frac{\lambda}{2} \|\mathbf{P}_i\|_2 + \frac{\lambda}{2} \|\mathbf{Q}_j\|_2 + \frac{\lambda}{2} \|\mathbf{w}\|_2 \\
 & + \frac{\lambda}{2} \|b_i\|_2 + \frac{\lambda}{2} \|b_j\|_2 + \frac{\lambda}{2} \|b_{ij}\|_2 \\
 & + \frac{\lambda}{2} \|\mathbf{\Lambda}\|_F^2 + \frac{\lambda}{2} \|\mathbf{\Gamma}\|_F^2 + \frac{\lambda}{2} \|\mathbf{V}\|_F^2
 \end{aligned} \tag{6.2}$$

where (i, j) is a pair of nodes, $L(\cdot)$ function now ranges from -1 (not link) to 1 (link), labels Y can be -1 (not link) or 1 (link), and $\phi_{ij} = 1$ if $y_{ij} = 1$ and $\phi_{ij} = p$ if $y_{ij} = -1$. Thus, Equation 6.2 should provide better estimates without increasing the overall training cost. As future work, we intend to propose an algorithm able to minimize this loss function. The main challenge will be to handle the exponential penalizations without numerical instabilities.

BIBLIOGRAPHY

ADAFRE, S. F.; RIJKE, M. de. Discovering missing links in Wikipedia. In: **Proc. 3rd International Workshop on Link Discovery**. ACM, 2005. (LinkKDD '05), p. 90–97. ISBN 1-59593-215-1. Available: <http://doi.acm.org/10.1145/1134271.1134284>. Cited 2 times on pages 18 and 38.

ADAMIC, L. A.; ADAR, E. Friends and neighbors on the web. **Social Networks**, v. 25, n. 3, p. 211 – 230, 2003. ISSN 0378-8733. Available: <http://www.sciencedirect.com/science/article/pii/S0378873303000091>. Cited on page 28.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **Knowledge and Data Engineering, IEEE Transactions on**, v. 17, n. 6, p. 734–749, June 2005. ISSN 1041-4347. Cited on page 28.

AYERS, P.; MATTEWS, C.; YATES, B. **How Wikipedia works: And how you can be a part of it**. [S.l.]: No Starch Press, 2008. Cited 2 times on pages 33 and 34.

BOTTOU, L. Stochastic gradient descent tricks. In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.-R. (Ed.). **Neural Networks: Tricks of the Trade (2nd ed.)**. Springer, 2012, (LNCS, v. 7700). p. 421–436. ISBN 978-3-642-35288-1. Available: <http://dblp.uni-trier.de/db/series/lncs/lncs7700.html#Bottou12>. Cited 2 times on pages 30 and 48.

CAI, Z.; ZHAO, K.; ZHU, K. Q.; WANG, H. Wikification via link co-occurrence. In: **Proc. 22nd ACM International Conference on Conference on Information and Knowledge Management**. ACM, 2013. (CIKM '13), p. 1087–1096. ISBN 978-1-4503-2263-8. Available: <http://doi.acm.org/10.1145/2505515.2505521>. Cited 2 times on pages 38 and 39.

CAMPBELL, M. J.; SWINSCOW, T. D. V. **Statistics at square one**. [S.l.]: John Wiley & Sons, 2011. Cited on page 62.

CHOI, W.; STVILIA, B. Web credibility assessment: Conceptualization, operationalization, variability, and models. **Journal of the Association for Information Science and Technology**, v. 66, n. 12, p. 2399–2414, 2015. ISSN 2330-1643. Available: <http://dx.doi.org/10.1002/asi.23543>. Cited on page 40.

CHU, W.; PARK, S.-T. Personalized recommendation on dynamic content using predictive bilinear models. In: **Proceedings of the 18th International Conference on World Wide Web**. New York, NY, USA: ACM, 2009. (WWW '09), p. 691–700. ISBN 978-1-60558-487-4. Available: <http://doi.acm.org/10.1145/1526709.1526802>. Cited on page 65.

CONDE, A.; LARRAÑAGA, M.; ARRUARTE, A.; ELORRIAGA, J. A.; ROTH, D. litewi: A combined term extraction and entity linking method for eliciting educational ontologies from textbooks. **Journal of the Association for Information Science and Technology**, v. 67, n. 2, p. 380–399, 2016. ISSN 2330-1643. Available: <http://dx.doi.org/10.1002/asi.23398>. Cited on page 18.

CONKLIN, J. Hypertext: An introduction and survey. **Computer**, v. 20, n. 9, p. 17–41, Sept 1987. ISSN 0018-9162. Cited on page 18.

DALIP, D. H.; GONCALVES, M. A.; CRISTO, M.; CALADO, P. Automatic quality assessment of content created collaboratively by web communities: a case study of Wikipedia. In: **Proceedings of the 2009 Joint International Conference on Digital libraries**. [S.l.: s.n.], 2009. p. 295–304. ISBN 978-1-60558-322-8. Cited on page 40.

_____. Automatic assessment of document quality in web collaborative digital libraries. **J. Data and Information Quality**, ACM, New York, NY, USA, v. 2, n. 3, p. 14:1–14:30, Dec. 2011. ISSN 1936-1955. Available: <<http://doi.acm.org/10.1145/2063504.2063507>>. Cited 4 times on pages 34, 40, 79, and 80.

_____. Exploiting user feedback to learn to rank answers in Q&A forums: A case study with Stack Overflow. In: **Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval**. [s.n.], 2013. p. 543–552. ISBN 978-1-4503-2034-4. Available: <<http://doi.acm.org/10.1145/2484028.2484072>>. Cited on page 40.

DALIP, D. H.; LIMA, H.; GONCALVES, M. A.; CRISTO, M.; CALADO, P. Quality assessment of collaborative content with minimal information. In: **Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries**. Piscataway, NJ, USA: IEEE Press, 2014. (JCDL '14), p. 201–210. ISBN 978-1-4799-5569-5. Available: <<http://dl.acm.org/citation.cfm?id=2740769.2740804>>. Cited 2 times on pages 79 and 80.

DILLON, A.; RICHARDSON, J.; MCKNIGHT, C. Navigation in hypertext: a critical review of the concept. In: JONES, D.; WINDER, R. (Ed.). **People and computers VII – (British Computer Society Conference Series: HCI'92)**. [S.l.]: Amsterdam: North Holland, 1990. Cited on page 18.

DONDIO, P.; BARRETT, S.; WEBER, S. Calculating the trustworthiness of a wikipedia article using dante methodology. In: **IADIS eSociety conference, Dublin, Ireland**. [S.l.: s.n.], 2006. Cited 2 times on pages 79 and 80.

DONDIO, P.; BARRETT, S.; WEBER, S.; SEIGNEUR, J. Extracting trust from domain analysis: A case study on the Wikipedia project. In: **Autonomic and Trusted Computing**. [s.n.], 2006. p. 362–373. Available: <http://dx.doi.org/10.1007/11839569_35>. Cited on page 40.

DONDIO, P.; BARRETT, S.; WEBER, S.; SEIGNEUR, J. M. Extracting trust from domain analysis: A case study on the wikipedia project. In: **Proceedings of the Third International Conference on Autonomic and Trusted Computing**. Berlin, Heidelberg: Springer-Verlag, 2006. (ATC'06), p. 362–373. ISBN 3-540-38619-X, 978-3-540-38619-3. Available: <http://dx.doi.org/10.1007/11839569_35>. Cited 2 times on pages 79 and 80.

DUNLAVY, D. M.; KOLDA, T. G.; ACAR, E. Temporal link prediction using matrix and tensor factorizations. **ACM Trans. Knowl. Discov. Data**, ACM, New York, NY, USA, v. 5, n. 2, p. 10:1–10:27, Feb. 2011. ISSN 1556-4681. Available: <<http://doi.acm.org/10.1145/1921632.1921636>>. Cited on page 28.

ERTEKIN, S.; RUDIN, C. On equivalence relationships between classification and ranking algorithms. **Journal of Machine Learning Research**, v. 12, p. 2905–2929, 2011. Available: <<http://jmlr.csail.mit.edu/papers/volume12/ertekin11a/ertekin11a.pdf>>. Cited on page 81.

FABER, R. **Why are reference works still important?** 2012. <<http://blog.oup.com/2012/09/why-are-reference-works-still-important/>>. Accessed: 2016-01-30. Cited on page 17.

FACELI, K.; LORENA, A.; GAMA, J.; CARVALHO, A. Inteligência artificial—uma abordagem de aprendizado de máquina. **Rio de Janeiro: LTC**, 2011. Cited on page 26.

FERREIRA, R.; PIMENTEL, M. d. G. C.; CRISTO, M. Exploring graph topology via matrix factorization to improve wikification. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2015. (SAC '15), p. 1099–1104. ISBN 978-1-4503-3196-8. Available: <<http://doi.acm.org/10.1145/2695664.2695930>>. Cited 2 times on pages 23 and 39.

GE, M.; HELFERT, M. A review of information quality research - develop a research agenda. In: ROBBERT, M. A.; O'HARE, R.; MARKUS, M. L.; KLEIN, B. D. (Ed.). **ICIQ**. MIT, 2007. p. 76–91. Available: <<http://dblp.uni-trier.de/db/conf/iq/iq2007.html#GeH07>>. Cited on page 40.

HAN, J.; CHEN, K.; WANG, J. Web article quality ranking based on web community knowledge. **Computing**, Springer, v. 97, n. 5, p. 509–537, 2015. Cited 2 times on pages 79 and 80.

HANADA, R.; CRISTO, M.; PIMENTEL, M. d. G. C. How do metrics of link analysis correlate to quality, relevance and popularity in wikipedia? In: **Proceedings of the 19th Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2013. (WebMedia '13), p. 105–112. ISBN 978-1-4503-2559-2. Available: <<http://doi.acm.org/10.1145/2526188.2526198>>. Cited 2 times on pages 22 and 40.

HASAN, M. A.; CHAOJI, V.; SALEM, S.; ZAKI, M. Link prediction using supervised learning. In: **SDM'06: Workshop on Link Analysis, Counter-terrorism and Security**. [S.l.: s.n.], 2006. Cited on page 39.

HU, M.; LIM, E.-P.; SUN, A.; LAUW, H. W.; VUONG, B.-Q. Measuring article quality in wikipedia: models and evaluation. In: **Proceedings of the sixteenth ACM Conference on information and knowledge management**. [s.n.], 2007. p. 243–252. ISBN 9781595938039. Available: <<http://dx.doi.org/10.1145/1321440.1321476>>. Cited on page 40.

HUANG, W. C.; TROTMAN, A.; GEVA, S. Experiments and evaluation of link discovery in the wikipedia. 2008. Cited on page 19.

KAPTEIN, R.; SERDYUKOV, P.; VRIES, A. D.; KAMPS, J. Entity ranking using wikipedia as a pivot. In: **Proceedings of the 19th ACM International Conference on Information and Knowledge Management**. New York, NY, USA: ACM, 2010. (CIKM '10), p. 69–78. ISBN 978-1-4503-0099-5. Available: <<http://doi.acm.org/10.1145/1871437.1871451>>. Cited on page 18.

KENDALL, M. G. A new measure of rank correlation. **Biometrika**, JSTOR, v. 30, n. 1/2, p. 81–93, 1938. Cited on page 56.

KIRTSIS, N.; STAMOU, S.; TZEKOU, P.; ZOTOS, N. Information uniqueness in wikipedia articles. In: **WEBIST (2)**. [S.l.: s.n.], 2010. p. 137–143. Cited on page 40.

KOREN, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: **Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. ACM, 2008. (KDD '08), p. 426–434. ISBN 978-1-60558-193-4. Available: <<http://doi.acm.org/10.1145/1401890.1401944>>. Cited 4 times on pages 21, 27, 39, and 48.

_____. Collaborative filtering with temporal dynamics. In: **Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2009. (KDD '09), p. 447–456. ISBN 978-1-60558-495-9. Available: <http://doi.acm.org/10.1145/1557019.1557072>. Cited 2 times on pages 21 and 27.

KORFIATIS, N.; POULOS, M.; BOKOS, G. Evaluating authoritative sources using social networks: An insight from wikipedia. **Online Information Review**, v. 30, n. 3, p. 252–262, 2006. Available: http://www.korfiatis.info/papers/OISJournal_final.pdf. Cited on page 40.

LAWRIE, D.; MAYFIELD, J.; MCNAMEE, P.; OARD, D. W. Cross-language person-entity linking from 20 languages. **Journal of the Association for Information Science and Technology**, v. 66, n. 6, p. 1106–1123, 2015. ISSN 2330-1643. Available: <http://dx.doi.org/10.1002/asi.23254>. Cited on page 80.

LI, C.; SUN, A.; DATTA, A. Tsdw: Two-stage word sense disambiguation using wikipedia. **Journal of the American Society for Information Science and Technology**, v. 64, n. 6, p. 1203–1223, 2013. ISSN 1532-2890. Available: <http://dx.doi.org/10.1002/asi.22829>. Cited 2 times on pages 18 and 80.

LI, W.-J.; YEUNG, D.-Y.; ZHANG, Z. Generalized latent factor models for social network analysis. In: WALSH, T. (Ed.). **IJCAI**. IJCAI/AAAI, 2011. p. 1705–1710. ISBN 978-1-57735-516-8. Available: <http://dblp.uni-trier.de/db/conf/ijcai/ijcai2011.html#LiYZ11>. Cited 2 times on pages 39 and 46.

LIBEN-NOWELL, D.; KLEINBERG, J. The link-prediction problem for social networks. **Journal of the American Society for Information Science and Technology**, Wiley Subscription Services, Inc., A Wiley Company, v. 58, n. 7, p. 1019–1031, 2007. ISSN 1532-2890. Available: <http://dx.doi.org/10.1002/asi.20591>. Cited 2 times on pages 28 and 39.

LING, C. X.; HUANG, J.; ZHANG, H. AUC: A Statistically Consistent and More Discriminating Measure Than Accuracy. In: **Proc. 18th International Joint Conference on Artificial Intelligence**. Morgan Kaufmann, 2003. (IJCAI'03), p. 519–524. Available: <http://dl.acm.org/citation.cfm?id=1630659.1630736>. Cited 2 times on pages 55 and 56.

Lü, L.; ZHOU, T. Link prediction in complex networks: A survey. **Physica A: Statistical Mechanics and its Applications**, v. 390, n. 6, p. 1150 – 1170, 2011. ISSN 0378-4371. Available: <http://www.sciencedirect.com/science/article/pii/S037843711000991X>. Cited on page 28.

MAIO, C. D.; FENZA, G.; LOIA, V.; PARENTE, M. Time aware knowledge extraction for microblog summarization on twitter. **Information Fusion**, Elsevier, v. 28, p. 60–74, 2016. Cited on page 79.

MALO, P.; SINHA, A.; WALLENIUS, J.; KORHONEN, P. Concept-based document classification using wikipedia and value function. **Journal of the American Society for Information Science and Technology**, Wiley Subscription Services, Inc., A Wiley Company, v. 62, n. 12, p. 2496–2511, 2011. ISSN 1532-2890. Available: <http://dx.doi.org/10.1002/asi.21596>. Cited on page 18.

MCALEESE, R. Navigation and browsing in hypertext. In: MCALEESE, R. (Ed.). **Hypertext: theory into practice**. [S.l.]: Intellect. Oxford, 1989. p. 6–44. Cited on page 18.

MENDES, P. N.; JAKOB, M.; GARCÍA-SILVA, A.; BIZER, C. Dbpedia spotlight: Shedding light on the web of documents. In: **Proceedings of the 7th International Conference on Semantic Systems**. New York, NY, USA: ACM, 2011. (I-Semantics '11), p. 1–8. ISBN 978-1-4503-0621-8. Available: <http://doi.acm.org/10.1145/2063518.2063519>. Cited on page 79.

MENON, A.; ELKAN, C. A log-linear model with latent features for dyadic prediction. In: **Data Mining (ICDM), 2010 IEEE 10th International Conference on**. [S.l.: s.n.], 2010. p. 364–373. ISSN 1550-4786. Cited 3 times on pages 27, 30, and 45.

MENON, A. K. **Latent Feature Models for Dyadic Prediction**. La Jolla, CA, USA: University of California at San Diego, 2013. AAI3557100. Cited on page 27.

MENON, A. K.; CHITRAPURA, K.-P.; GARG, S.; AGARWAL, D.; KOTA, N. Response prediction using collaborative filtering with hierarchies and side-information. In: **ACM. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2011. p. 141–149. Cited on page 39.

MENON, A. K.; ELKAN, C. Link prediction via matrix factorization. In: **Proc. 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II**. Springer-Verlag, 2011. (ECML PKDD'11), p. 437–452. ISBN 978-3-642-23782-9. Available: <http://dl.acm.org/citation.cfm?id=2034117.2034146>. Cited 6 times on pages 21, 27, 28, 39, 46, and 52.

MIHALCEA, R.; CSOMAI, A. Wikify!: Linking documents to encyclopedic knowledge. In: **Proc. Sixteenth ACM Conference on Conference on Information and Knowledge Management**. ACM, 2007. (CIKM '07), p. 233–242. ISBN 978-1-59593-803-9. Available: <http://doi.acm.org/10.1145/1321440.1321475>. Cited 4 times on pages 19, 20, 37, and 49.

MILNE, D.; WITTEN, I. H. Learning to link with Wikipedia. In: **Proc. 17th ACM Conference on Information and Knowledge Management**. ACM, 2008. (CIKM '08), p. 509–518. ISBN 978-1-59593-991-3. Available: <http://doi.acm.org/10.1145/1458082.1458150>. Cited 7 times on pages 18, 19, 20, 37, 50, 51, and 59.

_____. An open-source toolkit for mining Wikipedia. **Artif. Intell.**, Elsevier Science Publishers Ltd., v. 194, p. 222–239, Jan. 2013. ISSN 0004-3702. Available: <http://dx.doi.org/10.1016/j.artint.2012.06.007>. Cited 4 times on pages 38, 39, 62, and 65.

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill Higher Education, 1997. ISBN 0070428077. Cited on page 26.

NEWMAN, M. E. J. Clustering and preferential attachment in growing networks. **Phys. Rev. E**, American Physical Society, v. 64, p. 025102, Jul 2001. Available: <http://link.aps.org/doi/10.1103/PhysRevE.64.025102>. Cited on page 28.

PARANJAPE, A.; WEST, R.; ZIA, L.; LESKOVEC, J. Improving website hyperlink structure using server logs. In: **Proceedings of the Ninth ACM International Conference on Web Search and Data Mining**. New York, NY, USA: ACM, 2016. (WSDM '16), p. 615–624. ISBN 978-1-4503-3716-8. Available: <http://doi.acm.org/10.1145/2835776.2835832>. Cited 7 times on pages 22, 40, 72, 73, 74, 78, and 80.

POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. **Journal of Machine Learning Technologies**, v. 2, n. 1, p. 37–63, 2011. Cited on page 55.

RASSBACH, L.; PINCOCK, T.; MINGUS, B. **Exploring the Feasibility of Automatically Rating Online Article Quality**. 2007. <<http://upload.wikimedia.org/wikipedia/wikimania2007/d/d3/RassbachPincockMingus07.pdf>>. Cited on page 40.

RATINOV, L.; ROTH, D.; DOWNEY, D.; ANDERSON, M. Local and global algorithms for disambiguation to Wikipedia. In: **Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1**. Association for Computational Linguistics, 2011. (HLT '11), p. 1375–1384. ISBN 978-1-932432-87-9. Available: <<http://dl.acm.org/citation.cfm?id=2002472.2002642>>. Cited 3 times on pages 19, 20, and 37.

RENDLE, S. Factorization machines with libFM. **ACM Trans. Intell. Syst. Technol.**, ACM, v. 3, n. 3, p. 57:1–57:22, May 2012. ISSN 2157-6904. Cited 2 times on pages 21 and 39.

RESNICK, P.; IACOVOU, N.; SUCHAK, M.; BERGSTROM, P.; RIEDL, J. Grouplens: An open architecture for collaborative filtering of netnews. In: **Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work**. New York, NY, USA: ACM, 1994. (CSCW '94), p. 175–186. ISBN 0-89791-689-1. Available: <<http://doi.acm.org/10.1145/192844.192905>>. Cited on page 28.

RÜMMELE, N.; ICHISE, R.; WERTHNER, H. Exploring supervised methods for temporal link prediction in heterogeneous social networks. In: **Proceedings of the 24th International Conference on World Wide Web**. New York, NY, USA: ACM, 2015. (WWW '15 Companion), p. 1363–1368. ISBN 978-1-4503-3473-0. Available: <<http://doi.acm.org/10.1145/2740908.2741697>>. Cited on page 28.

STVILIA, B.; GASSER, L.; TWIDALE, M. B.; SMITH, L. C. A framework for information quality assessment. **Journal of the American Society for Information Science and Technology**, v. 58, n. 12, p. 1720–1733, 2007. Available: <<http://dblp.uni-trier.de/db/journals/jasis/jasis58.html#StviliaGTS07>>. Cited on page 40.

STVILIA, B.; TWIDALE, M. B.; SMITH, L. C.; GASSER, L. Information quality work organization in wikipedia. **Journal of the American Society for Information Science and Technology**, Wiley Subscription Services, Inc., A Wiley Company, v. 59, n. 6, p. 983–1001, 2008. ISSN 1532-2890. Available: <<http://dx.doi.org/10.1002/asi.20813>>. Cited on page 40.

SUNERCAN, O.; BIRTURK, A. Wikipedia missing link discovery: A comparative study. In: **AAAI Spring Symposium on Linked Data Meets Artificial Intelligence (Linked AI 2010)**, ser. **AAAI Spring Symposium, AS Symposium**, Ed., Stanford, USA. [S.l.: s.n.], 2010. Cited 2 times on pages 22 and 40.

TEJAY, G.; DHILLON, G.; CHIN, A. G. Data quality dimensions for information systems security: A theoretical exposition. In: **Security Management, Integrity, and Internal Control in Information Systems**. [S.l.]: Springer, 2006. p. 21–39. Cited on page 40.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. **International Journal of Data Warehousing and Mining (IJDWM)**, p. "1–13", 2007. Cited on page 27.

TSUNAKAWA, T.; ARAYA, M.; KAJI, H. Enriching Wikipedia's intra-language links by their cross-language transfer. In: **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics**. [S.l.: s.n.], 2014. p. 1260–1268. Cited on page 80.

TZEKOU, P.; STAMOU, S.; KIRTSIS, N.; ZOTOS, N. Quality assessment of wikipedia external links. In: **Proceedings of the 7th International Conference on Web Information Systems and Technologies**. [S.l.: s.n.], 2011. p. 248–254. ISBN 978-989-8425-51-5. Cited on page 40.

WANG, R. Y.; STRONG, D. M. Beyond accuracy: What data quality means to data consumers. **Journal of Management Information Systems**, v. 12, n. 4, p. 5–33, 1996. Available: <http://dx.doi.org/10.1080/07421222.1996.11518099>. Cited on page 40.

WEST, R.; PARANJAPPE, A.; LESKOVEC, J. Mining missing hyperlinks from human navigation traces: A case study of wikipedia. In: **Proceedings of the 24th International Conference on World Wide Web**. New York, NY, USA: ACM, 2015. (WWW '15), p. 1242–1252. ISBN 978-1-4503-3469-3. Available: <http://doi.acm.org/10.1145/2736277.2741666>. Cited on page 79.

WEST, R.; PRECUP, D.; PINEAU, J. Completing Wikipedia's hyperlink structure through dimensionality reduction. In: **Proc. 18th ACM Conference on Information and Knowledge Management**. ACM, 2009. (CIKM '09), p. 1097–1106. ISBN 978-1-60558-512-3. Available: <http://doi.acm.org/10.1145/1645953.1646093>. Cited 4 times on pages 19, 20, 38, and 39.

WIKIPEDIA. **Editorial oversight and control**. 2016. https://en.wikipedia.org/wiki/Wikipedia:Editorial_oversight_and_control. Accessed: 2016-02-13. Cited 2 times on pages 18 and 32.

_____. **Wikipedia entry**. 2016. <https://en.wikipedia.org/wiki/Wikipedia>. Accessed: 2016-02-13. Cited 2 times on pages 18 and 32.

_____. **Wikipedia Size comparisons**. 2016. https://en.wikipedia.org/wiki/Wikipedia:Size_comparisons. Accessed: 2016-02-13. Cited on page 32.

WILKINSON, R.; SMEATON, A. F. Automatic link generation. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 31, n. 4es, Dec. 1999. ISSN 0360-0300. Available: <http://doi.acm.org/10.1145/345966.346024>. Cited on page 19.

WITTEN, I. H.; FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques**. 3rd. ed. Morgan Kaufmann, 2011. Available: <http://amazon.com/o/ASIN/1558605525/>. Cited on page 61.

XU, Y.; LUO, T. Measuring article quality in wikipedia: Lexical clue model. In: **Proceedings of the 3rd Symposium on Web Society**. [S.l.: s.n.], 2011. p. 141 –146. ISSN 2158-6985. Cited on page 40.

YANG, S.-H.; LONG, B.; SMOLA, A.; SADAGOPAN, N.; ZHENG, Z.; ZHA, H. Like like alike: Joint friendship and interest propagation in social networks. In: **Proceedings of the 20th International Conference on World Wide Web**. New York, NY, USA: ACM, 2011. (WWW '11), p. 537–546. ISBN 978-1-4503-0632-4. Available: <http://doi.acm.org/10.1145/1963405.1963481>. Cited on page 65.

ZLATIC, V.; BOZICEVIC, M.; STEFANCIC, H.; DOMAZET, M. Wikipedias: Collaborative web-based encyclopedias as complex networks. **Phys. Rev. E**, American Physical Society, v. 74, p. 016115, Jul 2006. Available: <http://link.aps.org/doi/10.1103/PhysRevE.74.016115>. Cited on page 46.