Universidade Federal do Amazonas
Instituto de Computação
Programa de Pós-Graduação em Informática

Recognition and Linking of Product Mentions in User-generated Contents

Henry Silva Vieira

Manaus – 2018

Henry Silva Vieira

Recognition and Linking of Product Mentions in User-generated Contents

Tese apresentada ao Programa de Pós-Graduação em Informática, PPGI, da Universidade Federal do Amazonas como requisito parcial para obtenção do grau de Doutor em Informática.

Orientador: Prof. Dr. Altigran Soares da Silva

Manaus – Amazonas
Setembro de 2018

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

V658r

Vieira, Henry Silva
Recognition and Linking of Product Mentions in User-generated Contents / Henry Silva Vieira. 2018
127 f.: il. color; 31 cm.

Orientador: Altigran Soares da Silva
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Opinion mining. 2. Extracting target entities. 3. Categorizing target entities. 4. Product recognition and linking. I. Silva, Altigran Soares da II. Universidade Federal do Amazonas III. Título

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UFAM

FOLHA DE APROVAÇÃO

"Recognition and Linking of Product Mentions in User-generated Contents"

HENRY SILVA VIEIRA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Altigran Soares da Silva - PRESIDENTE

Prof. Edleno Silva de Moura - MEMBRO INTERNO

Pereira Calado

Prof. Pável Pereira Calado - MEMBRO EXTERNO

Leandro Balby Marinho

Prof. Leandro Balby Marinho - MEMBRO EXTERNO

Mirella Moro

Profa. Mirella Moura Moro - MEMBRO EXTERNO

Manaus, 25 de Setembro de 2018

À minha esposa Lúcia,

ao meu filho Caio.

Agradecimentos

A minha esposa pelo apoio incondicional, mesmo nas horas mais difíceis. Ao meu filho pela compreensão de minha ausência.

Aos meus pais que sempre me apoiaram durante toda a vida de estudos. Obrigado por todos os incentivos que me fizeram chegar onde estou.

Ao meu orientador, Altigran Soares, por sempre acreditar em mim. Obrigado pelos direcionamentos e horas de dedicação. Ao Pável Calado, por acolher a mim e minha família em Lisboa, por seus conselhos e ajuda.

A FPF Tech pela liberação e apoio financeiro para o desenvolvimento deste trabalho. A FAPEAM e CAPES pelas bolsas de estudo.

Ao amigo Edson César pelo pequeno empurrão para fora da porta. Aos que participaram de alguma forma na realização desta tese.

Obrigados a todos!

Once upon a time not far away
We all swore the oath to seize the day
Moving on at the speed of light
Still united, undivided

We're brothers in arms searching for gold and glory
Stay true to your heart, come follow, heed the call
But nothing's forever, no nothing at all

Come inside, true templars of the world
Raise your voice, let's speak for the unheard
On and on the story must go on
Running free we always will be

We are what we are and we always strive for glory
Together as one we shine like the sun
Beyond the divine we're soaring

Shining on so glorious, the bloodline of true warriors
Fighting for the right to be; part of the legacy

Searching for the rainbow's end
The gold awaits for you my friend
Courage is your guiding star
And it will take you far

Shining on so glorious, the bloodline of true warriors
Fighting for the right to be; part of the legacy

Searching for the rainbow's end
The gold awaits for you my friend
Courage is your guiding star
And it will take you far

HammerFall, Origins

Abstract

Online social media has grown into an essential part of our daily life. Through these media, users exchange information that they generate by using many different communication mechanisms. In this context, more and more users pass on and trust information published by other users on a large variety of topics, including opinion and information about products. Automatically extracting and processing user-generated information in social media can provide relevant information and knowledge to a variety of interesting applications. In particular, one of the content analysis techniques most often applied to social media is that of opinion mining. One of the basic tasks associated with opinion mining is extracting and categorizing target entities, i.e., identifying entity mentions in text, and linking these entity mentions to unique real world entities about which the opinions are made. In our work, we focus on target entities of a specific, and currently relevant, type: consumer electronic products. Such products are the main subject of opinions posted by users on a number of posts in discussion forums and retail sites over the Web. In this work, we are interested in using the unstructured textual content generated by social media users to continuously allow enriching the knowledge about products represented in product catalogs. Therefore, the task we address here is how to recognize and link mentions to products in user generated textual content to the product, from a catalog, they refer to. We claim that two basic sub-tasks arise: first, extraction of target entities mentions from unstructured textual content; second, disambiguation of extracted entities, i.e., linking extracted mentions to their real world counterpart. In this work, we developed methods to address these two sub-tasks. This thesis details these tasks, discusses our ideas for the methods we developed, and presents our contributions and results towards this goal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online social media, such as blogs, microblogs, collaborative encyclopedias, social networks and discussion forums, has been experiencing an astonishing growth, not only in terms of content volume but also in popularity and social impact. Through these media, users exchange information that they generate by using many different communication mechanisms that are accessible and scalable [Kaplan and Haenlein, 2010]. These media are characterized by diverse content, which is generated by ordinary users, and by a rich set of interaction possibilities between content generators and content consumers [Pang and Lee, 2008]. User-generated information may be shared, discussed, commented, transformed, "liked", cited, etc. Using such interaction mechanisms, users publish information they generate in different formats such as reviews, social network messages, forum posts and their replies, etc.

In this context, more and more users pass on and trust information published by other users on a large variety of topics, including opinion and information about products. Thus, user-generated information is very important because of its potential influence on consumer decisions [Presi et al., 2014, Sethna et al., 2017]. People are able to make informed decisions based on information gathered from social media content. According to the Wall Street Journal [Penn and Zalesne, 2013], 92% of users trust more the information related to products and services published in social media by regular users than on the information published in other sources, such as advertisement. Such behavior has also been reported in the literature [Choi and Lee, 2017]. Even among people who shop outside the Web, a substantial portion (about 51%) say they make decisions based on online user-generated information [Moghaddam and Ester, 2013]. Also, the relevance of comments produced by online communities is demonstrated by a 20% gain in shopping conversion at online retailers when a site publishes users reviews [Moghaddam and Ester, 2013].

Automatic processing and extracting user-generated information and user interactions from social media can provide relevant information and knowledge to a variety of interesting applications [Feldman, 2013, Liu, 2012, Breck and Cardie, 2017], such as: (a) predicting user group behavior; (b) recommending more reliably and with higher quality; (c) pricing of products and services where the interests of suppliers and consumers are maximized; (d) summarizing user group opinions; and (e) calculating the return of investment of certain advertisement content, product or service. Furthermore, this user-generated information being publicly available enables to automatically estimate, among other things, the general user sentiment polarity (negative, positive or neutral) and their emotions towards different products; and enrich product catalogs which contains only manufacturer provided data.

As a consequence, the analysis of the contents produced by users in these media is an important and urgent matter. In particular, one of the content analysis techniques most often applied to social media is that of *opinion mining*. Opinion mining is concerned with the extraction of people's senti-

> *"I can't decide between a* ***Galaxy S3*** *or a* ***Iphone 5****, so I think I am going to buy BOTH."*
>
> *"I would be surprised if* ***iP5*** *came so soon, but the antenna issue needs to be correctly addressed."*
>
> *"Can you use an unlocked at&t* ***iphone5*** *on verizon's network?"*
>
> *"I have a good condition* ***Apple iPhone 5*** *(Verizon, 16GB, White) for sale."*
>
> *"It's true there will always been a need for something more powerful - I won't be throwing out my* ***Panasonic LX-5*** *or my* ***Canon 7D*** *- they have their uses - but I find myself taking an awful lot of pictures with the* ***iPhone 4s****."*

Figure 1.1: Mentions to products, using different surface forms, in an online forum.

ments, moods, attitudes and emotions towards some entity or its aspects [Liu, 2012, Feldman, 2013]. This task is particularly appealing when carried out over comments and reviews made by custumers and buyers on consumer products [Castellanos et al., 2011, Feldman, 2013, Santosh et al., 2016, Poria et al., 2016, Chawla et al., 2017].

One of the basic tasks associated with opinion mining is that of *extracting and categorizing target entities* [Liu, 2012], i.e., identifying entity mentions in text, and linking these entity mentions to unique real world entities about which the opinions are made. In general, these entities can be people, brands, companies, places, etc.

In our work, we focus on target entities of a specific, and relevant, type: consumer electronic products, such as smartphones, digital cameras and Bluray players. Such products are the subject of opinions posted by users on a number of posts in discussion forums and retail sites over the Web.

Figure 1.1 illustrates the complexity of recognizing and linking target consumer electronic products (entities) with some real world examples. In the figure, each sentence is an excerpt of a user's post on Howard Forums – an influential Web site that hosts forums related to mobile phones, with over one million members[1]. Each mention to a product is highlighted in bold.

Not surprisingly, in these sentences, product mentions are very ambiguous, since users typically reference the same product using many different *surface forms* or *entity expressions* [Liu, 2007]. For instance, the *Apple iPhone 5* smartphone is mentioned using four different surface forms: *Iphone 5*, *iP5*, *iphone5* and *Apple iPhone 5*. To add to the difficulty, products of categories outside those discussed in the forum are also mentioned (in this case, the digital cameras “Panasonic LX-5” and “Canon 7D”). Furthermore, textual content in social media is usually informally written and not free of misspellings.

The problem of identifying and linking products mentions in user-generated contents has motivated several research initiatives. Zhang [Zhang and Liu, 2011], for example, studied the problem of mining brands and product names from forum posts, with the ultimate goal of identifying opinions on entities of a same type. In the CPROD1 contest [Melli and Romming, 2012], candidates were asked to develop methods that recognized consumer product mentions in user-generated Web content. The method that obtained the best results, described in [Wu et al., 2012], is based on a combination of techniques, such as a simple grep-like matching, a rule-based technique, and two supervised conditional random fields (CRF) models. The authors in [Yao and Sun, 2016] propose a method called GREN to recognize mobile phone names in posts from Web forums. The method starts by first generating candidate names from forum text. These candidate names capture variations of mobile phones names. A CRF is then used to predict whether a candidate name actually refers to a phone model. The CRF model is trained from a set of sentences obtained in a semi-automatic manner with manual labeling effort.

In this work, we are interested in using the unstructured textual content

[1] `http://www.howardforums.com/forums.php`

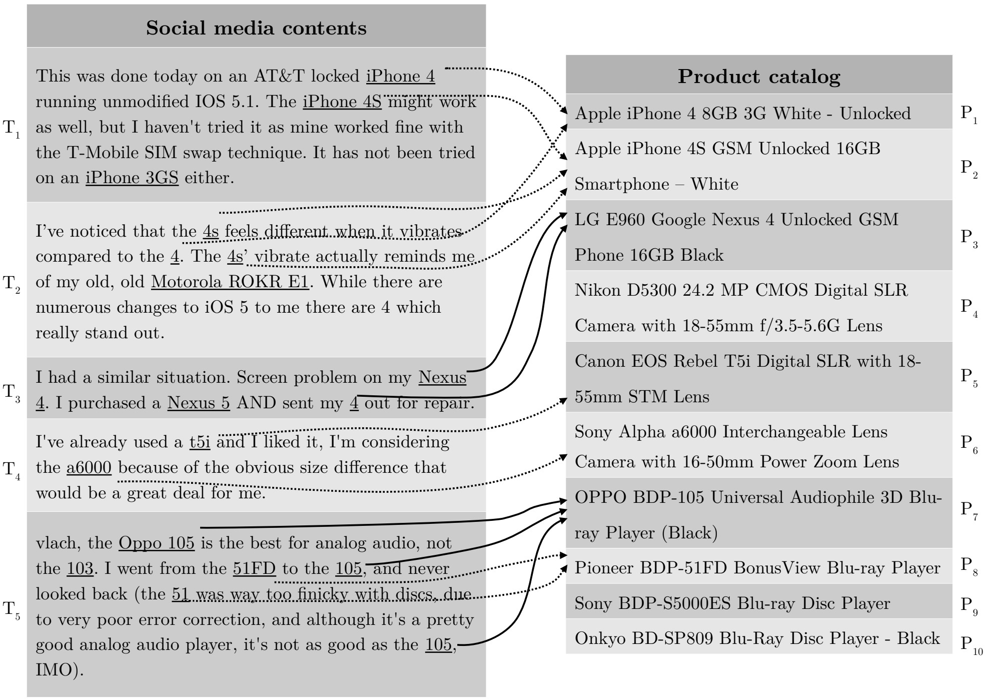
generated by social media users to continuously allow enriching the knowledge about products represented in product catalogs. Therefore, the task we address here is *how to recognize and link mentions in user generated textual content to the product, from a catalog, they refer to.*

1.1 Product Recognition and Linking

We approach the *Product Recognition and Linking* task as two basic sub-tasks. Product Recognition is the process of automatically identifying a mention m to a product in a text document or fragment. Product Linking is the process of automatically associating product mention m to an entry representing that product in a catalog. Given a textual product mention m, the unstructured text t in which it appears, and a product catalog C, the goal is to establish a link from m to its corresponding real world product $p \in C$.

Figure 1.2 illustrates real world examples to demonstrate the sub-tasks. The left column represents text excerpts from user-generated information posts on Web forums (T_1 to T_5), and the right column represents entries from a sample product catalog (P_1 to P_{10}).

We regard the first sub-task as an instance of the more general named entity recognition (NER) task [Sarawagi, 2008], where the problem is to recognize entity mentions in natural language text. State-of-the-art NER techniques are based on probabilistic graphical models, e.g., conditional random fields (CRF) [Sarawagi, 2008]. Although effective, probabilistic graphical models for NER are difficult to directly apply to the problem we focus on here mainly because these methods require a large number of representative labeled data for training. Labeled data is costly to acquire, and this cost is even higher if we consider that there are many distinct product categories, each one with particularities in terms of lexico-semantic characteristics [Zhang and



Social media contents

T1: This was done today on an AT&T locked iPhone 4 running unmodified IOS 5.1. The iPhone 4S might work as well, but I haven't tried it as mine worked fine with the T-Mobile SIM swap technique. It has not been tried on an iPhone 3GS either.

T2: I've noticed that the 4s feels different when it vibrates compared to the 4. The 4s' vibrate actually reminds me of my old, old Motorola ROKR E1. While there are numerous changes to iOS 5 to me there are 4 which really stand out.

T3: I had a similar situation. Screen problem on my Nexus 4. I purchased a Nexus 5 AND sent my 4 out for repair.

T4: I've already used a t5i and I liked it, I'm considering the a6000 because of the obvious size difference that would be a great deal for me.

T5: vlach, the Oppo 105 is the best for analog audio, not the 103. I went from the 51FD to the 105, and never looked back (the 51 was way too finicky with discs, due to very poor error correction, and although it's a pretty good analog audio player, it's not as good as the 105, IMO).

Product catalog	
Apple iPhone 4 8GB 3G White - Unlocked	P1
Apple iPhone 4S GSM Unlocked 16GB Smartphone – White	P2
LG E960 Google Nexus 4 Unlocked GSM Phone 16GB Black	P3
Nikon D5300 24.2 MP CMOS Digital SLR Camera with 18-55mm f/3.5-5.6G Lens	P4
Canon EOS Rebel T5i Digital SLR with 18-55mm STM Lens	P5
Sony Alpha a6000 Interchangeable Lens Camera with 16-50mm Power Zoom Lens	P6
OPPO BDP-105 Universal Audiophile 3D Blu-ray Player (Black)	P7
Pioneer BDP-51FD BonusView Blu-ray Player	P8
Sony BDP-S5000ES Blu-ray Disc Player	P9
Onkyo BD-SP809 Blu-Ray Disc Player - Black	P10

Figure 1.2: Examples of the Product Linking task and related sub-task.

Liu, 2011].

In Figure 1.2, each product mention is underlined to illustrate the output of our first sub-task, i.e., the recognition of target product mentions. Thus, in this example, there are 18 product mentions to be recognized. From Figure 1.2, we can note that the recognition process must be robust enough to correctly address cases of ambiguity that occur quite often.

Ambiguity is a particularly hard problem in social media content, since users typically reference the same entity using many different *surface forms* or *entity expressions* [Liu, 2007], within the same text or across different texts [Feldman, 2013]. For instance, take excerpt T_3 presented in Figure 1.2. In the example, a successful recognition technique should be able to correctly capture all variations of the *LG Nexus 4*, .i.e, *Nexus 4* and *4*. Moreover, such technique should be able to differentiate between occurrences of the number

"4" and the ambiguous product mentions "4". For instance, in T_2 one can find two occurrences of the string "4". The first one, which is underlined, corresponds to a product mention, while second, which is not underlined, does not. Only by considering the context in which the strings occur it is possible to devise strategies able to correctly differentiate between usages of the string "4".

In social media, unstructured informal text is prevalent. This textual content is in natural language; usually with misspellings and informally written [Eisenstein, 2013]. An additional complication arises from the fact that new products (and surface forms) appear very often. Under these conditions, continually providing an adequate volume of representative training instances or hand-crafted recognition rules is an unfeasible task to carry out manually.

Once product mentions have been correctly recognized, the next step is to link these mentions to their corresponding products from a catalog. The second sub-task is similar to the problem of Entity Linking [Rao et al., 2013, Shen et al., 2012, Cucerzan, 2007, Ji et al., 2017]. However, in our case the target is a product catalog instead of a knowledge base, such as Wikipedia, which is used in many of the previous works in the literature (e.g., [Cucerzan, 2007, Ceccarelli et al., 2013]). This is a significant diference since, contrary to what happens in product catalogs, knowledge bases are usually rich in contextual information for each entity they represent. For instance, in the case of Wikipedia, pieces of contextual information that have been exploited by methods in the literature [Ceccarelli et al., 2013, Li et al., 2013, Rao et al., 2013, Shen et al., 2012, Zhang et al., 2011] are: (a) entity page textual content and length; (b) link graph structure; (c) reference link textual content; and (d) infoboxes. Our problem scenario can rely only on a small number of contextual information from offers from a product catalog.

Traditionally, linking methods target entities of types person, organization or locations, but not products. Targeting products is motivated by the volume of worldwide consumer e-commerce sales. A forecast presented in [eMarketer, 2014] states a growth of 20.1% in sales to a total of 1.500 trillion dollars in 2014. Furthermore, mentioning products is a very common practice in Web forums. In a sample of more than 60,000 posts from AVS Forum, about half of the posts have product mentions.

Eventually, one could consider the use of Wikipedia as the linking target for products. However, the problem addressed here is different from the entity linking problem. The input in the entity linking problem is a Knowledge Base, usually the Wikipedia, containing contextual information about entities, while the input in the product linking problem presents only the product description (or product title), from a product catalog. Thus, there is much less information available in the product linking problem to resolve ambiguities. This difference prevents us to take advantage from previously proposed entity linking solutions. Any comparison between methods to solve these distinct problem would also be unfair.

In Figure 1.2, an arrow from a mention depicts the link to the corresponding product in the catalog. This corresponds the second sub-task of linking recognized mentions to the product they refer to in the catalog. From Figure 1.2, we can note that different surface forms refer to the same product, such as *51FD* and *51* (T_5) that refer to the *Pioneer BDP-51FD* (P_8). On the other hand, a same surface form *4* (T_2 and T_3) refers to different products, *Apple iPhone 4* (P_3) and *LG Nexus 4* (P_3). Also some product mentions, such as the *Motorola ROKR E1* (T_2), don't have a corresponding product in the catalog.

This second sub-task has its own form of ambiguity. In this case, the ambiguity is between product mentions sharing the same surface form. For instance, take the excerpts T_2 and T_3 presented in Figure 1.2. The particularly common surface form *4* can easily be linked to the *Apple iPhone 4*, *Apple iPhone 4s* or *LG Nexus 4* by a naive linking strategy. Only by considering the context in which the mention occurs it is possible to devise strategies able to correctly link mentions such as *4* to its respective product.

Finally, user-generated information usually has a limited local context. Even human readers would have difficulty while reading such information without further inspecting the whole context in which the information was posted. Again, take the excerpt T_2. Without inspecting previous related posts or forum thread titles a user might confuse such mention as relating to the *Apple iPhone 4s* or the *LG Nexus 4*.

1.2 Main Contributions

In Chapter 3 we present the first contribution. We present a novel distantly supervised method, called *ModSpot*, for learning a CRF model to undertake the task of identifying product model numbers occurring in a set of sentences extracted from forum posts given as input. For enabling the learning process, the method requires only a set of seed model numbers examples in the same category, which means it does not require that annotated training sentences from the target forum are provided.

In Chapter 4 we present a second contribution to the sub-task of recognizing product mentions. Although distantly supervised and requiring only a set of seed examples, our first contribution to the sub-task of recognizing product mentions from unstructured textual content was limited to product model numbers. To overcome this limitation, we proposed a new method, called *ProdSpot*. We demonstrate that it is possible to build a high-performing product recognition system which labels product mentions from user-generated content by taking as input only unstructured product descriptions, a list of brands, and the target user-generated content. In particular, this is achieved without any manually labeled data by bootstrapping a supervised classifier using a set of examples of product surface forms extracted from the product descriptions. These example surface forms are extracted in an unsupervised manner from product descriptions leveraging only the list of brands.

In Chapter 5 we present our contribution to the second sub-task of linking recognized mentions to their real world counterpart. This contribution presents a method, called *ProdLink*, to link product mentions to their respective real-world products. We argue that this problem can be effectively solved using a set of evidences that can be extracted from the social media content and product descriptions. Specifically, we show which features should be used, how they can be extracted, and then how to combine them through machine learning techniques. ProdLink is an supervised end-to-end solution for product linking, capable of both recognizing product mentions in natural language text from public forum posts and of linking those mentions the entries in a catalog.

1.3 Thesis Organization

This thesis is organized as follows. In Chapter 2 we review related approaches previously presented in the literature to the general tasks of opinion mining, entity recognition and entity linking. We also discuss the more specific works related to product linking and normalization. In Chapter 3 we present the details and steps of our first contribution toward the goal of product linking. In Chapter 4 we present the details and steps of our product mention recognition contribution. In Chapter 5 we present the details and steps of our contribution for product linking. For each contribution, we also present experimental results for verifying the effectiveness of our approach. Chapter 6 concludes this thesis and presents future works.

Chapter 2

Related Work and Background

In this chapter, we review related approaches to the sub-tasks of *extracting and categorizing target entities*, i.e., named entity recognition and entity linking. We also discuss specific works related to product mention recognition, and product linking and normalization. We begin by presenting an overview of opinion mining, and its relation to the sub-tasks we address in this work.

2.1 Opinion Mining

Opinion mining is the application of techniques from natural language processing (NLP), text classification, and machine learning to extract people's sentiments, moods, attitudes and emotions towards some specific entity or its aspects [Liu, 2012, Feldman, 2013]. Opinion mining is mainly applied at three levels of granularity: (a) document; (b) sentence; or (c) entity and aspect. Aspects are the components, attributes, or features of an entity. For example, the picture quality of a digital camera.

Document level, the simplest form of opinion mining, targets the sentiment classification of entire documents. At this level, it is assumed that the document contains opinion towards one single entity, e.g., given a product review, the objective is to determine if the sentiment it contains is positive or negative towards the product. At a greater granularity, sentence level opinion mining performs sentiment classification for each sentence in a document. This allows a detailed view of the different sentiments expressed in documents. It is assumed that the opinion target for each sentence is known, and sentence are previously classified as subjective or objective. Only sentences classified as subjective are further analyzed.

The two previous levels of sentiment classification work well when either the document or each sentence refers to a single known entity. However, in many cases, user-generated content is full of entities that have many different aspects, and users may have different sentiments towards each one of the entities and its aspects. Entity and aspect level opinion mining aim at sentiment classification towards an specific opinion target aspect and entity, both identified from text.

At the entity and aspect level, an opinion is defined as a quintuple (e_i, a_{ij}, s_{ijkl}, h_k, t_l), where e_i is an entity, a_{ij} is an aspect of e_i, s_{ijkl} is the sentiment towards aspect a_{ij}, h_k is the opinion holder, and t_l is the time when the opinion is expressed [Liu, 2012]. Sentiment s_{ijkl} is positive, negative or neutral; and may have varied intensity levels.

This definition provides a framework where structured data (quintuple) is identified from unstructured text. The definition also yields the basic tasks associated with entity and aspect opinion mining, i.e., extraction and categorization of each quintuple component from user-generated content.

In our work, we focus on the task of *extracting and categorizing target* *entities*, and its two sub-tasks: (a) identifying entity mentions in text; and (b) linking entity mentions into unique entities about which the opinions are made.

We focus on target entities of a specific, and currently relevant, type: consumer electronic products, such as smartphones, digital cameras and Blu-ray players. Such products are the main subject of opinions posted by users on a number of posts in discussion forums and retail sites over the Web.

We regard the first sub-task as an instance of the more general named entity recognition task [Sarawagi, 2008], where the problem is to recognize entity mentions in natural language text. While the second sub-task is similar to the problem of Entity Linking [Rao et al., 2013, Shen et al., 2012, Cucerzan, 2007, Ji et al., 2017]. However, in our case the target is a product catalog instead of a knowledge base, such as Wikipedia.

2.2 Named Entity Recognition

Named entity recognition is a very common sub-task of *information extraction* that aims at identifying *named entities* in unstructured text [Sarawagi, 2008, Hobbs and Riloff, 2010, Navigli, 2009, Derczynski et al., 2017, Peng and Dredze, 2015]. A named entity is a real-world object, such as people, geographic locations, organizations, or products, that can be denoted with a proper name [Grishman and Sundheim, 1996], and are typically noun phrases comprised of one to a few tokens in unstructured text.

Early NER systems used hand-crafted patterns and rules defined by human experts for performing recognition. The authors in [Rau, 1991] presented a algorithm that automatically extracts company names from financial news. It generates the most likely variations of company names that are used along with a set of manually built recognition rules. Another rule based system is proposed in [Sekine and Nobata, 2004], where the authors describe the creation of dictionaries and an automatic tagger based on pattern rules for named entities in Japanese.

As manual devising rules became tedious and recognition systems targeted more noisy unstructured textual sources (where rules were found to be too brittle), statistical and machine learning methods started to become more prominent [Sarawagi, 2008]. The authors in [Florian et al., 2003] present a ensemble framework using four diverse classifiers (robust linear classifier, maximum entropy, transformation-based learning, and hidden Markov model) to perform named entity recognition. Most of the participants in the CoNLL-2003 NER shared task employed a wide variety of statistical and machine learning techniques [Tjong Kim Sang and De Meulder, 2003].

Statistical methods of entity recognition generally convert the recognition task to a problem of treating unstructured text as a sequence of tokens and the recognition problem is to assign labels to each token. Take a sequence of tokens $\mathbf{x} = x_1, \ldots, x_n$. During label assignment, each x_i has to be classified as one label in $\mathcal{Y}$ giving a sequence of tags $\mathbf{y} = y_1, \ldots, y_n$. The label set $\mathcal{Y}$ constitute the set of entity types (e.g., people, geographic locations, organizations, or products) and a special label "other" that represents tokens that do not belong to any of the entity types. This token labeling can be seen as a generalization of single-token classification called *sequence classification*. State-of-the-art sequence classification techniques are based on probabilistic graphical models, e.g., conditional random fields (CRF) [Sarawagi, 2008].

2.2.1 Conditional Random Fields

Conditional random fields (CRF) [Lafferty et al., 2001] are a type of discriminative probabilistic classifier used for sequence classification, i.e., jointly labeling sequences of tokens. A CRF[1] model is a form of undirected graphical model that defines a distribution over label sequences $\mathbf{y}$ given a particular observation sequence $\mathbf{x}$. As an undirected graphical model, or Markov random field, a CRF may have an arbitrary graphical structure, provided it represents the conditional independences in the label sequences being modeled, however, when modeling sequences, the simplest and most common graph structure is that of a first-order linear chain.

P($\mathbf{y}|\mathbf{x}$) in a CRF model is defined as:

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}) =& \frac{1}{Z(\mathbf{x})} \exp(\sum_{i=1}^{n} (\sum_{k} \lambda_k f_k(y_i, y_{i-1}, \mathbf{x})) + (\sum_{\ell} \mu_\ell g_\ell(y_i, \mathbf{x}))) \\ =& \frac{1}{Z(\mathbf{x})} \exp(\sum_{i=1}^{n} (\lambda^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}) + \mu^T \mathbf{g}(y_i, \mathbf{x}))) \end{aligned}$$

where

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp(\sum_{i=1}^{n} (\lambda^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}) + \mu^T \mathbf{g}(y_i, \mathbf{x})))$$

and $\mathbf{f}$,$\mathbf{g}$ are, respectively, feature functions relating label pairs and feature functions derived from observed tokens. Values $\lambda_1, \lambda_2, \lambda_3, \ldots, \mu_1, \mu_2, \mu_3, \ldots$ are model parameters which indicate the weight attributed to each feature function.

Training a CRF model is performed over a set of labeled training sequences where the objective is to determine the values of $\lambda_1, \lambda_2, \lambda_3, \ldots, \mu_1, \mu_2, \mu_3, \ldots$ that maximize P($\mathbf{y}|\mathbf{x}$) for such training examples.

Inference, i.e., labeling sequences, is performed with model parameters λ e μ, where a label sequence $\mathbf{y}^*$ is found by maximizing P($\mathbf{y}|\mathbf{x}$):

$$\mathbf{y}^* = \arg\max_{y} \exp(\sum_{i=1}^{n} (\lambda^T \mathbf{f}(y_i, y_{i-1}, \mathbf{x}) + \mu^T \mathbf{g}(y_i, \mathbf{x}))).$$

[1] Only for the sake of clarity through the text, we use CRF as a synonym for CRF model.

2.2.2 Named Entity Recognition in Opinion Mining

The problem of named entity recognition as a task of recognizing target entities in opinion mining text has been recently investigated in several papers in the literature [Wu et al., 2012, Zhang and Liu, 2011, Jakob and Gurevych, 2010].

The authors in [Jakob and Gurevych, 2010] focus on extracting entities that are subjects of opinion, as part of an opinion mining task. They tackle the problem using a CRF model, trained with data from different domains, such as movies, web-services, cars and cameras. Each trained domain model is tested both within the same domain and across other domains, reportedly outperforming the baseline in both scenarios. Although successful, their method follows as traditional supervised approach. This implies the need for previously labeled data, which is not necessary in this work.

An approach for mining brands and product names from forum posts is presented in [Zhang and Liu, 2011], where the motivation is to find opinions on entities of the same type. The authors regard this task as similar to NER, and model it as set expansion problem. More specifically, their method starts with a set of seed entities and tries to expand it with other similar entities. Set expansion is achieved through *Bayesian Sets*, an algorithm that estimates the probability of a candidate entity being of the same class as the existing seeds.

Several methods have been proposed to compete in the CPROD1 contest [Melli and Romming, 2012], where candidates were asked to develop solutions for recognizing and disambiguating product mentions in user generated Web content. The method that obtained the best results, described in [Wu et al., 2012], is based on the combination of several distinct recognition models. The first model is a simple grep-like matching, based on the annotated input. For the second model, the authors use a rule-based technique, where a token is considered a product mention when all the generated rules apply. Those rules check conditions such as (1) the occurrence of tokens with a specific character sequence; (2) the presence of semantic patterns, such as a pronoun followed by a product mention; and (3) if the token belongs to an exclusion list containing dictionary terms, stop words, capitalized nouns, and abbreviations. Finally, two CRF models previously trained for the task of identifying product mentions are used. The two models are trained using different sets of features. Despite providing accurate results, the method is based on the availability of resources such as rules, templates, positive and negative dictionaries, etc., which may restrict its application when such information is not present. Our proposal, although equally based on a CRF model, does not depend on manually constructed rules or any type of supervised input.

Yao et al. proposed a method named GREN to recognize mobile phone mentions from Web forums [Yao and Sun, 2016]. However, instead of directly recognizing mentions from sentences as in most NER methods, they propose an approach where candidate tokens are classified as being a true mobile phone mention or not. The method assumes as input a collection of mobile phone formal, i.e., official, names and posts from a Web forum that is know to contain mentions to mobile phones. Candidates are generated by filtering noun phrases based on rules related to variations of a brand name, and the occurrence of tokens from the same Brown cluster as the tokens in a mobile phone formal name. The Brown algorithm is an unsupervised method that generates word clusters from unlabeled text [Brown et al., 1992]. Variations of a brand name are identified, by a set of 4 rules, from Brown clusters containing a brand token. It is not clear how the filtering and brand name variation rules generalize to other product domains. Candidate classification is done by a CRF model trained from a set of sentences obtained in a semi-automatic manner with manual labeling effort. As the method relies on Brown clusters and mobile phone formal names to identify candidates, it is unable to recognize mobile phone mentions not provided as input.

In a broader perspective, previous work in the literature have exploited the use of bootstrap techniques and semi-supervised learning in NER tasks.

In [Liao and Veeramachaneni, 2009], the authors propose a semi-supervised learning algorithm using a CRF. The method starts with a supervised input to train a CRF model which recognizes entity mentions of type location, organization, and person in financial news. Their method is applied to a large unlabeled corpus from which new training data is derived, based on the model's classification confidence. At each iteration, the classifier is trained with data from previous step and then used to label yet unlabeled data, based on specific rules for each entity type. Nevertheless, this work is still different from our method as it requires an initial labeled input. Also, it focuses on financial news, a completely distinct scenario compared to the one addressed here.

2.2.3 Distant Supervision and Noisy Labels in NER

The standard supervised machine learning approach consists in learning a classifier model from fully labeled data. Labels are, usually, manually assigned to each example and are expected to be correct, i.e., the data does not contain false-positive or false-negative examples. The classifier model, learned from the manually labeled data, is then used to predict the labels of new unseen samples.

In our case, the supervised machine learning approach is based on a probabilistic graphical model, e.g., CRF. Although effective, probabilistic graphical models for NER are difficult to directly apply to the problem we focus on here mainly because these methods require a large number of representative labeled data for training. Labeled data is costly to acquire, and this cost is even higher if we consider that there are many distinct product categories, each one with particularities in terms of lexico-semantic characteristics [Zhang and Liu, 2011].

The method described in [Teixeira et al., 2011] starts with a set of (non-annotated) news items and a dictionary of names frequently found in the news. First, the method annotates names in the set of news items by considering capitalized matches with entries in the dictionary. It then uses the matched sentences to compose the seed corpus, from which the seed corpus is then used to infer a CRF, iteratively applying it to the seed corpus to increase the number of completely annotated sentences.

The work in [Putthividhya and Hu, 2011] focuses on mining short product listing titles, to extract product attributes. The authors formulate the *product attribute extraction problem* as a NER task and investigate supervised and semi-supervised approaches. Specific strategies are proposed for extracting special attributes, such as brands. This problem is clearly different from the one addressed in our work. However, one of the strategies adopted to extract brands is quite close to our approach. Specifically, the authors propose a semi-supervised method that expands a given initial list of brands (the seed dictionary) and discovers new brand names from eBay listing data. A set of seed values is used to automatically generate labeled training data for a CRF model. Also, for the specific case of brand discovery, this initial seed list must contain only names that are unambiguously brands. Thus, the authors remove ambiguous seeds from input before matching. The training and test data is generated by matching n-gram tokens in listing titles to all the entries in the brand seed dictionary. Our work, on the other hand, identifies product references in free text, and our target input is characteristically ambiguous —

discarding ambiguous seeds is not an option.

Our product recognition systems are trained without manually labeled data. Instead, training data is obtained through *distant supervision*, where a set of examples are automatically annotated. Since there is no manual intervention in the annotation process, the set of examples obtained is both noisy and non-exhaustive. Noisy because not all automatically annotated examples are correctly labeled as product or as non-product. Non-exhaustive because the input to the process is unable to output all the possible products mention examples (and variations). Thus, we are dealing with what is commonly called *noisy training data*.

The literature usually distinguishes two types of training data errors: *feature noise* and *label noise* [Zhu and Wu, 2004, Frénay and Verleysen, 2014]. Feature noise affects the observed attributes of examples, while label noise affects the observed labels (classes) assigned to the examples. Our work must deal primarily with the second case. In fact, mislabeling is regarded as potentially more harmful to classification than feature noise, since (1) the number of features per example is much greater that the number of labels (usually one) per example, and (2) the importance of each feature for classification is usually diluted between all features in the learned model.

There are three main approaches to handle label noise [Frénay and Verleysen, 2014]: (1) use learning algorithms that are robust to label noise; (2) improve the quality of the training data by filtering training examples before training occurs; and (3) use learning algorithms that directly model label noise during training. In the work presented here, we take the filtering approach, by removing from the training data examples that are likely to be false-negatives or false-positives.

2.2.4 Synthetic Training Examples

Supervised classifiers usually require a large amount of labeled data for training. If the labeled data contains an appropriate amount of examples, the model will be sufficiently close to the target distribution, resulting in successful generalization to unseen instances. Thus, the number of training examples is a key issue to successful generalization. When fewer data is available, incorporating prior knowledge is an alternative way of enabling classifiers to generalize, even if trained on a small data sample. One way to incorporate prior knowledge is by generating new examples from the available data. These are called *virtual* or *synthetic* training examples and their creation can be formalized as follows [Niyogi et al., 1998].

Take a transformation T such that if an instance/label pair $(\mathbf{x}, y)$ is a valid example, then $(T\mathbf{x}, y)$ is also a valid example. Given a set of n examples $D = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ and a transformation T, we generate the set of synthetic examples $D' = (\mathbf{x}'_1, y_1), \ldots, (\mathbf{x}'_n, y_n)$ such that $\mathbf{x}'_i = T\mathbf{x}_i$.

The transformation T is defined as to encode prior knowledge, that will be represented in the generated examples.

For example, in [Song et al., 2004] the authors apply the synthetic examples framework to the task of NER in the biomedical domain. They expand the training data by exploiting the fact that the syntactic role of a named entity is a noun and the basic syntactic structure of a sentence is preserved if a noun is replaced by another noun. Their results demonstrate that, after using synthetic examples, both the precision and recall levels are increased when compared to a dataset without the expansion.

2.3 Entity Linking

Our second sub-task, linking recognized mentions to the product they refer to in a catalog, resembles the task of *entity linking*, which has attracted much attention in the literature [Cucerzan, 2007, Dredze et al., 2010, Zhang et al., 2011, Ceccarelli et al., 2013, Rao et al., 2013, Li et al., 2013, Ji et al., 2017]. Entity linking consists in finding mentions to named entities in text and linking them to the corresponding entity in a given knowledge base, usually Wikipedia. The main difference to product linking is that such knowledge bases are rich in contextual information, which can be exploited to disambiguate the entity mentions. In most cases, these methods make intensive use of unique features from Wikipedia. Such unique information includes links between related entities, rich textual information describing each entity, and link anchor text. Also, entity linking tasks assume that, for each mentioned entity there is only one corresponding entry in the knowledge base, whereas, in our case, a single mention can relate to a whole set of products (e.g., the mention *iphone* can be linked to all the iPhone models).

In a seminal work, Cucerzan [Cucerzan, 2007] proposed a method to disambiguate named entities in a text document T by matching the contextual information extracted from T and from Wikipedia entity pages, using the vector space model. The method also leverages the category associated with the target entity (e.g., *person*). In [Dredze et al., 2010], the authors propose a supervised learning approach that relies on features such as Wikipedia link graph structure, entity-page categories and keywords. To reduce manual training, Zhang et al. [Zhang et al., 2011] propose to automatically expand acronyms into full entity names, generating training data from the resulting unambiguous mentions. In addition, a topic model is used to capture semantic features between the document and the target knowledge base. Later, several other machine learning based methods have been proposed for the Entity Linking problem. They exploit features such as entity-page titles, co-citations and anchor texts [Ceccarelli et al., 2013], graph statistics and category structure [Rao et al., 2013], *redirect pages* and *disambiguation pages* [Shen et al., 2012]. In addition to features from Wikipedia, [Li et al., 2013] uses features from entity-related Web pages retrieved using Google.

However, none of the above solutions can be directly applied to the problem of product linking. In product linking the only input available are product descriptions (or product titles) from a product catalog, and the user-generated text. This represents much less information, and thus less exploitable features, than that available on a more formal knowledge base. Although the techniques used might be regarded as similar, different features of the available data must be exploited to discover and disambiguate product mentions.

A related, but distinct task, is that of *record linkage* [Elmagarmid et al., 2007]. Record linkage aims at matching records that refer to the same entity across different datasets. Each record describing an entity contains a set of attribute values used during linkage with the assumption that duplicate records should have equal or similar attribute values. Differently, in the entity linking task, entities mentions which needs to be linked are present in the unstructured text and the mentions do not have associated attribute values [Shen et al., 2015].

2.4 Previous Work on Product Recognition and Linking

Product recognition and linking was the main subject of the CPROD1 contest, held within ICDM 2012 [Melli and Romming, 2012]. In this contest, candidates were asked to develop methods that recognized consumer product mentions in user generated Web content. Each recognized mention should be linked to the corresponding set of products in a product catalog. The catalog contained records that represented purchasable consumer products from the electronics and automotive categories. The available data also included text items with manually labeled product mentions, to support supervised mention recognition. Each candidate submission was ranked based on the average F_1 score for the union of predicted and correctly disambiguated product mentions.

The solution that obtained the best results in the contest is described in [Wu et al., 2012]. Its first step is to recognize the product mentions in the input texts. This is accomplished by combining distinct recognition models. The first model is a simple grep-like matching scheme based on the annotated input. In a second model, the authors use a rule-based technique, where a token is considered a product mention if a set of manually generated rules apply. The rules check conditions such as the occurrence of tokens with a specific character sequence (e.g. "iPhone"), the presence of semantic patterns, such as a pronoun followed by a product mention, and if the token does not belong to an exclusion list containing dictionary terms, stop words, capitalized nouns, and abbreviations. Finally, two supervised conditional random field (CRF) models are applied. The second step is the linking task itself. For this, the authors proposed a simple strategy based on substring matching, selecting entries from the catalog based solely on the occurrence of the mention within the product title.

Although [Wu et al., 2012] achieved a good performance on CPROD1, there is clearly room for improvement. Its major drawback comes from the fact that many products share common substrings. Take, as an example, the sentence *"I've noticed that the 4s feels different when it vibrates compared to the 4."* from a post in the Howard Forums website, and consider that the product catalog is comprised of cell phones. If we consider only substring grep-like matching, the particularly common surface form *4* will be linked to, at least, the following products: *Apple iPhone 4*, *Apple iPhone 4S*, *LG Nexus 4*, and *Samsung Galaxy S4*.

Another highly related work is that of [Yao and Sun, 2016]. The authors propose a method to recognize mobile phone mentions in posts from Web forums. Their aim is then to *normalize* mentions that refer to a same product. The normalization consists in linking mentions to their *canonical* or *formal* phone names, given as input. The method assumes that formal names are structured records containing a *brand*, a *model name* and *model number*.

The authors start by recognizing product mentions in posts as phone name variations, using a CRF model. Their solution then carries out the product linking task, which, in this case, consists of linking the mentions to the formal name records, by adopting a two-step rule-based approach. In the first step, a given product mention is linked to a specific formal name if all the characters in the mention are contained in the brand and model name and if they are arranged in the same order. This match is case insensitive and considers that Roman numbers match their Arabic counterpart. In the second step, their method tries to link the remaining product mentions, i.e. those that were not linked in the first step, by matching its tokens to the tokens previously associated with the products. After this second step, some mentions might be linked to multiple candidate products. To solve this, a mention is linked only to the candidate product whose formal name has co-occurred most frequently with it, in the input posts.

Similarly to the work of [Wu et al., 2012], however, this method is also prone to errors for the most commonly ambiguous product mentions. For instance, it also fails in handling the mention *4* that, depending on the context, should be linked to *Apple iPhone 4* or *LG Nexus 4*. Moreover, it assumes that formal names are properly segmented in *brand*, *model name* and *model number*. Although these may be appropriate for the author's goals, it may not be effective in a more general product linking scenario. As discussed in [Melli, 2014], the segmentation problem can be challenging by itself, which may limit the application of the method.

A related, but distinct task, is addressed in [Dalvi et al., 2009a] and [Dalvi et al., 2009b]. The authors consider the problem of review matching. Given a list of structured objects and a review text, the aim is to identify the object from the list that is the topic of the review. The authors assume that each object has a set of structured attributes that describes it, e.g., location and cuisine for restaurants. These works are different from ours as their goal is to match a whole review text while we are interested in linking mentions on the text to a product. Namely, in our forum scenario, the text from a single post may contain mentions to several different products.

Chapter 3

Recognizing Product Model Numbers

In this chapter we present the details and steps of our first contribution. This is a preliminary contribution toward the goal of product mention recognition. Although effective, this contribution was generalized in the method described in Chapter 4.

The main task we focus on is recognizing model numbers of products which are mentioned in user posts. The key requirement is that products must be *relevant*, i.e., they must be of a category specified by the user in the input.

We propose a novel method, called ModSpot[1], for learning a CRF to undertake the task of identifying products model numbers occurring in a set of sentences extracted from forum posts given as input. For enabling the learning process, it requires only a set of seed model numbers in the same category, which means it does not require that annotated training sentences from the target forum are provided. Notice that in ModSpot, the category is implicitly determined by the provided seeds; to recognize Blu-ray player

[1] Product **Mod**el Number **Spot**ter.

*"As I mentionend in my **BDP-93** review, Oppo has created quite a name for itself with their Blu-ray players for home theater devotees. ... So here we have Oppo's brand new **BDP-103**. ..."*

*"Got my **103** and everything looking good. ... with my Sony **NX30** camcoder, the date ... Same file played with the **93** and no ..."*

*"Is there a way to bypass the scaler on the **D2**? If I feed the **D2** 1080p24 (assuming BluRay DVD eos this) will it output unprocessed 1080p24? ..."*

Figure 3.1: Products referred to by model numbers in reviews.

mentions, the seeds provided must consistently refer to Blu-ray players. We argue that obtaining these sets of seeds is fairly easy, since they are available in product listings from retail Web sites or from public data repositories such as Wikipedia or Freebase. We observe that in reviews users very often refer to a particular electronic product by means of its *model number*[2]. Consider the excerpt from a review posted at Amazon.com presented in Figure 3.1(a). In this example, two distinct Blu-ray players are mentioned by their respective model numbers, "BDP-93" and "BDP-103". Notice that the model numbers are the only way of distinguishing these two products in this review.

In many cases, mentions of product model numbers are very ambiguous, since users typically reference the same product model using many different *surface forms* or *entity expressions* [Liu, 2012], within the same text or across different texts [Feldman, 2013]. For instance, take the excerpt from the AVS Forum[3] presented in Figure 3.1(b). In the example, the same two Blu-ray players are mentioned using their respective surface forms "93" and "103". Additionally, a product out of the Blu-ray player category is also mentioned ("NX30"). As another example of ambiguity in this problem, Figure 3.1(c) presents another post from AVS in which an audio/video receiver ("D2") is

[2] In here, we adopted the same jargon of retail stores, in which "model number" refers to a code that identifies a particular product model. This code, though, is not necessarily a number.

[3] `http://www.avsforum.com/` – an influential Web site that hosts forums on electronic equipment with over one million members.

mentioned twice. In this example, the term “1080p24” does not correspond to a product model number, although it could be regarded as so by a naive strategy, e.g., one that uses regular expressions.

In a nutshell, ModSpot has two main steps. In the first step, it performs a bootstrapping process, where input seeds are expanded into multiple surface forms to account for variations. Each expanded surface form is annotated in input sentences to train an initial CRF. In the second step, a self-training [Yarowsky, 1995] process is carried out. ModSpot uses the output of the initial CRF to discover new model numbers in unlabeled sentences. New model numbers with high probability are added to the set of seeds and are again expanded into multiple surface forms, that are again annotated in unlabeled input sentences to train a new CRF. This process runs until no new seeds are found.

Experiments in four different settings demonstrate that ModSpot achieves similar or better results compared to using supervised CRF with the same feature set. Additionally, our method converges at around 9-14 iterations, where there is no growth in the seed set. All the experimented settings exhibit higher F_1 and the number of seeds is about 40% larger by the end of the process. In particular, the expansion in seeds helps to achieve higher recall levels

The main advantage of ModSpot is that, given only a set of seeds (examples of product model numbers), it generates a CRF able to recognize other product model numbers from this category in posts from distinct forums, without requiring previous annotations in posts from the target forum. For instance, in our experiments, we show that the same seed set in the category of Blu-ray players was used to identify Blu-ray players in posts from two different forums, one in English and the other in Portuguese. In this experiment, no previous annotated posts from any of the forums were necessary.

3.1 Overview

The method we propose is based on the self-training framework [Yarowsky, 1995], normally used to wrap complex models for semi-supervised learning. It consists of four steps: (1) train a classifier using a labeled set; (2) estimate the labels in a larger unlabeled set; (3) add the instances predicted with larger confidence to the training set; and (4) repeat the process. In our algorithm, we repeat steps 1 to 3 until no new instances that can be added to the training set are found. In our case, step 1 is replaced by a bootstrapping strategy where the labeled set is automatically annotated using examples of product model numbers, and step 3 is replaced by an strategy that identify new examples of product model numbers to be used during bootstrapping.

This framework suits our needs because, in the problem we tackle here, supervised data is costly to acquire, and this cost is even higher if we consider many distinct consumer electronic product categories and domains. Thus, our algorithm makes extensive use of unlabeled data which, in our setting, is abundant. We exploit this characteristic by estimating labels for the unlabeled data, so that it can be used for training a CRF. This simple strategy has two drawbacks: incorrect labeled instances can be included in the training set and errors are reinforced. To cope with these problems, we ensure reliable labeling by requiring that the input for the CRF training meets specific recognition criteria. In our self-training setting, we observe that the probabilities of the instances converge such that a final CRF is obtained after a number of iterations.

In addition to the self-training framework, our method includes a bootstrapping step, whose goal is to make the process independent from the particular target forum. Specifically, it take as input a set of *seeds*, i.e., examples of product model numbers, to automatically generate an initial training set of labeled sentences. To maximize the number of sentences in this initial training set, we also detect variations, i.e., distinct surface forms, of the given seeds that may appear in input posts. Notice that the examples of product model numbers we take as seeds can be easily obtained from crawling product lists from on-line retail Web sites other open sources such as Wikipedia and Freebase.

3.2 ModSpot

We detail our method in Algorithm 1. Let S_0 be an initial set of seeds, that is, examples of product model numbers, and U be a set of unlabeled sentences extracted from posts of a target forum. An initial training set L is automatically generated by bootstrapping from U (Lines 1-2). In this bootstrapping process, we detect surface form variations using our *SFDetection* algorithm explained in Section 3.3. This detection should account for the various ways users typically mention product models. Product model mention variations are automatically annotated in sentences from U to generate training set L. Non-annotated sentences are assigned to set T. This set will be later used later to enhance the seeds set with newly discovered seeds using a linear-chain CRF.

Next, an initial CRF $\hat{\theta}_0$ is trained using the automatically annotated sentences in L (Line 3). CRF training is performed with stochastic gradient descent and L1 regularization. Now, with a bootstrapped CRF, our self-training iteration process (Lines 5-13) begins. The algorithm iterates until it converges to a state where output from the trained CRFs does not change from one iteration to the next. In Lines 6-9, the algorithm performs the label

Algorithm 1 ModSpot

```
Input: Set of seeds $S_0$, set of unlabeled sentences $U$
 1: $L \leftarrow$ SFDetection($U$, $S_0$)                    ▷ Bootstraps and detects Surface Forms
 2: $T \leftarrow U - L$
 3: Build the initial CRF $\hat{\theta}_0$ from $L$ only
 4: $i \leftarrow 1$                                          ▷ Self-training Process
 5: repeat
 6:     Use $\hat{\theta}_{i-1}$ to label unlabeled sentences in $T$    ▷ Use CRF to predict new labels
 7:     $C \leftarrow$ the set of sentences labeled by $\hat{\theta}_{i-1}$
 8:     $M \leftarrow$ SeedExpansion($C$)                     ▷ Selects likely seeds
 9:     $S_i \leftarrow S_{i-1} \cup M$
10:     $L \leftarrow$ SFDetection($U$, $S_i$)                ▷ Detects Surface Forms
11:     $T \leftarrow U - L$
12:     Build a new CRF $\hat{\theta}_i$ from $L$ only        ▷ Re-train CRF with new labels
13: until $|S_i| = |S_{i-1}|$                                 ▷ No new seeds are found
14: return $\hat{\theta}_i$                                   ▷ Return last CRF generated
```

prediction step to discover new likely seeds. First, the current CRF $\hat{\theta}$ labels the unlabeled sentences in T, creating a labeled sentence set C. From the sentences in C, we run our *SeedExpansion* step that discovers new seeds into set C. Our SeedExpansion step is detailed in Section 3.4. Finally, set M is added to the current seeds set S_i (Line 9) expanding the initial seeds set with newly discovered product model mentions.

Between the label prediction and the CRF re-training is another bootstrapping process (Lines 10-11). This process is the same that automatically generated the training set L during initialization, but uses the expanded seeds set S_i as input. Again product model mention variations are automatically annotated in sentences from U to generate the training set, and each non-annotated sentence is added to T.

In Line 12, the algorithm trains a new CRF, which is the final step in our method. This step estimates the CRF $\hat{\theta}_i$ parameters using the automatically annotated sentences in L generated from the bootstrapping process executed after the label prediction step.

Our self-training algorithm convergence is determined by the difference of the seed set S_i from the current iteration and the seed set from the previous iteration S_{i-1} (Line 13). This condition guarantees that the algorithm stops, and outputs the last CRF generated during the process execution.

3.3 Surface Forms Detection

Users typically employ surface form variations while mentioning product model numbers in Web forums. To account for these variations, we propose a surface form detection algorithm.

Let s be a seed from a set S containing examples of product model numbers. We model s as a sequence $x_1, x_2, \ldots, x_n$, where each x_i is a token composed of only letters or only digits. Each token x_i is called a *block*. Thus, s is a sequence of blocks. As an example, take product model number "BDP-51FD" given as an input seed. Its sequence of blocks is "BDP", "51", "FD". We define a *surface form* f of s as being a sequence of blocks such that one of the conditions below holds.

- Condition 1: f is a sub-sequence $f_1, f_2, \ldots, f_n$ of s, with $n > 1$, occurring in at least one input sentence, or
- Condition 2: f is a single block of s, composed by digits only, occurring in at least one input sentence, and the context in which f occurs in input sentences is *similar* to the context in which some known surface form of s occurs in the input sentences.

For instance, according to the first condition, possible surface forms of "BDP-51FD" are "BDP51FD"; "BDP51"; and "51FD", if they occur in at least one sentence. In the case of the second condition, "51" is a possible surface, given that the context in which it occurs in the input sentences is similar to that of another occurrence of s.

The second condition is very important for the correct detection of surface forms. In fact, in our experiments we found that about one third of the surface forms are composed by digits only. Thus, to avoid confusing any number occurring in sentences as a surface form, we use the context implied by the input sentence for disambiguation. For this, we use a popular word-sense disambiguation strategy based on cosine similarity [Navigli, 2009].

Precisely, we consider as context portions of terms occurring before and after a surface form. Then, the context similarity is computed as follows. Consider a surface form t, which satisfies our first condition, represented by a vector $\mathbf{v}$ in which v_i is the frequency of term v_i in t within a fixed size context in the same input sentence. Also consider the same vector representation $\mathbf{w}$ for a candidate surface form t_c. We define the similarity between t_c and t as:

$$sim(\mathbf{w}, \mathbf{v}) = \frac{\mathbf{w} \cdot \mathbf{v}}{|\mathbf{w}||\mathbf{v}|} = \frac{\sum_{i=1}^{m} w_i v_i}{\sqrt{\sum_{i=1}^{m} w_i^2}\sqrt{\sum_{i=1}^{m} v_i^2}} \tag{3.1}$$

where m is the size of a common vocabulary used by $\mathbf{v}$ and $\mathbf{w}$.

We consider the two contexts as being similar if $sim(\mathbf{w}, \mathbf{v})$ is above a predefined threshold. We arbitrarily determined a value of 0.5 for this threshold. In experiments whose results are reported in Section 3.5, we validated this choice. In addition, after a few initial experiments, not reported here, we concluded that a context of size 3 is suitable for our application.

We describe our detection process in Algorithm 2. Our surface form detection algorithm is comprised of two sequential iterations. The first iteration detects surface form variations from the first condition (Lines 2-7), while the second iteration (Lines 8-13) detects variations based on the second condition. Each iteration loops through all sentences $u \in U$ and detects surface forms that satisfy any of the two conditions. If a given condition is satisfied, the algorithm annotates surface form f; the annotated sentence u in which f was

Algorithm 2 SFDetection

```
Input: Set of seeds $S$, set of unlabeled sentences $U$
 1: $L \leftarrow \emptyset$
 2: for $i \in 1, 2$ do
 3:     for each sentence $u \in U$ do
 4:         for each surface form $f$ in $u$ using Condition $i$ do
 5:             label $f$ in $u$
 6:             $L \leftarrow L \cup u$
 7:         end for
 8:     end for
 9: end for
10: return $L$
```

detected is added to L (Lines 5 and 11) to serve as training data for the CRF. We recall that detecting surface forms based on Condition 2 is only possible after surface forms have been detected using Condition 1.

3.4 Seed Expansion

We are interested in seeds that are likely product model numbers to expand our seeds sets. The automatic seed expansion process must be carried out without adding spurious seeds to the seeds sets. Otherwise, it would introduce and propagate errors during our self-training process to the CRFs as seeds are used to automatically annotate training sentences. To propagate fewer errors, our method adopts strict criteria in order to use only high confidence seeds.

Our first criterion is CRF labeling confidence based on the probability of label assignment. Traditionally, inference using CRF is computed using the Viterbi algorithm [Lafferty et al., 2001]. Given a set of feature functions $f_1, f_2, \ldots, f_n$, the Viterbi decoding finds the most likely sequence assignment $\mathbf{y} = (y_1, y_2, \ldots, y_t)$, called the Viterbi path, given an observation sequence $\mathbf{x} = (x_1, x_2, \ldots, x_t)$, defined as $\mathbf{y}^* = \text{argmax}_y p(\mathbf{y}|\mathbf{x})$. In our case, an observation sequence $\mathbf{x}$ derives from a sentence from set C. To be computationally effective, the Viterbi algorithm avoids searching all possibilities of $\mathbf{y}$ by storing the probability of the most likely path that accounts for x_i and ends in state

y_j at time i. The recursive formula is:

$$\delta_{i+1}(y_j) = \max_{y' \in \mathbf{y}} \left[\delta_i(y') \text{exp} \left(\sum_{k=1}^{n} \lambda_k f_k(y', y_j, \mathbf{x}, i) \right) \right] \tag{3.2}$$

where λ_k is a learned weight for each CRF feature function $f_k(y', y_j, \mathbf{x}, i)$.

Although computationally efficient, the Viterbi algorithm output path score is not normalized thus cannot be interpreted as a probability. This is a major shortcoming for our self-training process as we are interested in high probability labels from set C. In other words, even if a term in a sentence in C is labeled as a product model mention, we only consider it if its labeling score is above a given probability threshold.

Thus, for obtaining normalized output scores, we use the so-called *posterior decoding* (Forward-Backward Algorithm) [Chen et al., 2008, Culotta and McCallum, 2004] instead of the classical Viterbi decoding. This algorithm allows CRF to output normalized scores by evaluating all possible paths given an observation. The Forward pass is defined by:

$$\alpha_{i+1}(y_j) = \sum_{y' \in \mathbf{y}} \left[\alpha_i(y') \text{exp} \left(\sum_{k=1}^{n} \lambda_k f_k(y', y_j, \mathbf{x}, i) \right) \right] \tag{3.3}$$

where α_{i+1} is the Forward-values vector used by the algorithm.

The Backward pass is defined by:

$$\beta_{i-1}(y_j) = \sum_{y' \in \mathbf{y}} \left[\beta_i(y') \text{exp} \left(\sum_{k=1}^{n} \lambda_k f_k(y', y_j, \mathbf{x}, i) \right) \right] \tag{3.4}$$

where β_{i-1} is the Backward-values vector used by the algorithm.

Note that Equation 3.3 and 3.4, instead of computing the maximum likely path as in Equation 3.2, all paths are considered.

As our goal is to use only high confidence seeds, we determined a high threshold value of 0.9 for our probability confidence. Terms labeled below this threshold in a sentence in C are discarded.

Finally, our second criterion is the number and type of blocks from the terms labeled by the CRF. Consider that a labeled term is also modeled as a sequence $x_1, x_2, \ldots, x_n$, where each x_i is a token composed of only letters or only digits, and each token x_i is a block. A valid seed has at least one block of each type, and the blocks have length greater than one.

The output from our seeds selection process is a new seeds set M that is added to the initial seeds set.

3.5 Experimental Results

In this section, we evaluate ModSpot using a variety of datasets on the tasks of product model number mention recognition. We first describe the datasets, the evaluation metrics and the baseline used. Then, we report the results on extraction quality and performance over all datasets.

3.5.1 Setup

We start by reporting the experimental datasets used throughout the experiments, the evaluation metrics and the supervised baseline used. Finally, we report the feature used by the models.

Experimental Data

We used four distinct datasets[4] from three different product categories of consumer electronics to evaluate our method. The categories are audio/video receivers (AVR), Blu-ray players (BDP) and LCD displays (LCD). Each dataset is a collection of sentences crawled during a two month period between January and February 2014 from one of two popular forums on the Web, namely, AVS Forum and HTFORUM. AVS Forum (AVS) is an influential Web site that hosts forums on electronic equipment with over one million members and

[4] Available at `http://shine.icomp.ufam.edu.br/~henry/datasets.html`

Dataset	Labeled Sentences	Product Model Mentions	Numeric Mentions	Posts with Mentions	Avg. Mentions per Post
AVS AVR	986	234	115 (49.1%)	99 (49.5%)	2.4
AVS BDP	1151	280	110 (39.3%)	96 (48.0%)	2.9
AVS LCD	963	135	31 (23.0%)	60 (30.0%)	2.2
HT BDP	875	148	42 (28.4%)	71 (35.5%)	2.0

Table 3.1: Datasets statistics – 200 posts per dataset.

more than 20 million posts. HTFORUM (HT) is the premier forum for audio/video enthusiasts in Portuguese, with about 5 million posts. This dataset was included specifically to verify the resilience of our method to different languages. From each product category and forum, we collected 100 threads and 10 pages per thread. By doing so, we were able to achieve a broad coverage of different product models in each category.

From each dataset, we randomly sampled 200 posts and manually labeled them to form our golden set. Table 3.1 gives some statistics for all the datasets used in our experiments. In all labeled sentences we found a large amount of product model mentions. For instance, in AVS/BDP alone there were 280 mentions. Numeric-only mentions represent a significant portion of user employed surface forms. These account for roughly half of the mentions in AVS/AVR. On average, the datasets have approximately 81 posts with mentions from the 200 sampled posts It is important to note that, in accordance with the task we address, only products in the specific category were labeled. Also, notice that we do not label cases where product mentions are made using pronouns, since we do not address the problem of anaphora resolution here.

The initial input seeds were crawled from Amazon.com for each product category. The surface forms were manually extracted from product descriptions, and they contain only each product's model such as BDP-S5100, BDP-51FD, etc. The amount of initial seeds for each category is 747 for AVR, 323 for BDP, and 1375 for LCD.

Evaluation Metrics

To evaluate our method, we used the well-known *precision*, *recall*, and F_1 metrics. Precision is the ratio of tokens correctly classified among all tokens predicted as composing a product mention. Recall is the ratio of tokens correctly classified among all tokens manually labeled as composing a product mention. F_1 is the harmonic mean of precision and recall.

More precisely, let G be the golden set with manually labeled tokens and S the result set yielded by our method. We define precision (P), recall (R) and F_1 as:

$$P = \frac{|G \cap S|}{|S|} \qquad R = \frac{|G \cap S|}{|G|} \qquad F_1 = 2 \times \frac{(P \times R)}{(P + R)} \tag{3.5}$$

Baseline

Our experiments compare ModSpot with a supervised CRF generated for each dataset. We consider supervised CRF to be a suitable baseline for comparison with our method, since it is regarded as very effective for NER tasks [Sarawagi, 2008]. We notice that the only methods we have found in the literature for the product mention recognition task were those proposed for the CPROD1 contest [Melli and Romming, 2012]. Unfortunately, the method which achieved the best results in the contest, [Wu et al., 2012], could not be used as a baseline, since it is based on resources (rules, templates, positive and negative dictionaries, etc.), that we were not able to obtain to properly reproduce its results. Nevertheless, this method is not easily applied to the scenario we address, because there are many distinct product categories and new products/surface forms appear very often. In such a scenario, the overhead for applying the method proposed in [Wu et al., 2012] seems to be too significant to make it viable. We used the CRF implementation presented in [Lavergne et al., 2010], trained with stochastic gradient descent and L1 regularization.

Set	Description
0	Current token
1	Tokens in a context window of size 3
2	Part-of-speech tag of the current token and of the tokens in the context window
3	Token begins with uppercase, token is all uppercase and token has a character that is uppercase
4	Token is numeric, token is a combination of alphanumeric characters and token has punctuation

Table 3.2: Features used by CRF.

Features

To build the CRF, we adopt a number of features widely used in previous work [Zhang and Liu, 2011, Jakob and Gurevych, 2010, Sarawagi, 2008]. Although CRFs are flexible enough to allow specific features for different domains, we used the same set of features and configurations in all experiments. It is important to note that our self-training procedure uses the same set of features and configurations as the baseline. These features are described in Table 3.2. We also opted not to use the BILOU representation [Ratinov and Roth, 2009], since most target product mentions are comprised of only one token.

The final feature set used in our method was validated using the forward selection strategy described in [Kohavi and John, 1997]. The feature set is initially populated with just one feature, and new features are gradually added to the CRF configuration, from more generic features, to features specific to our domain. Each feature set is evaluated with a CRF using 10-fold cross-validation.

Figure 3.2 gives the forward selection results for each dataset. Except for the BDP category, each set of features gradually raises F_1. Lower precision results are compensated by growth in recall. The final set of features gives the highest F_1 on all datasets mainly because of high-recall. After the validation process, we concluded that all chosen features were suitable for the task.

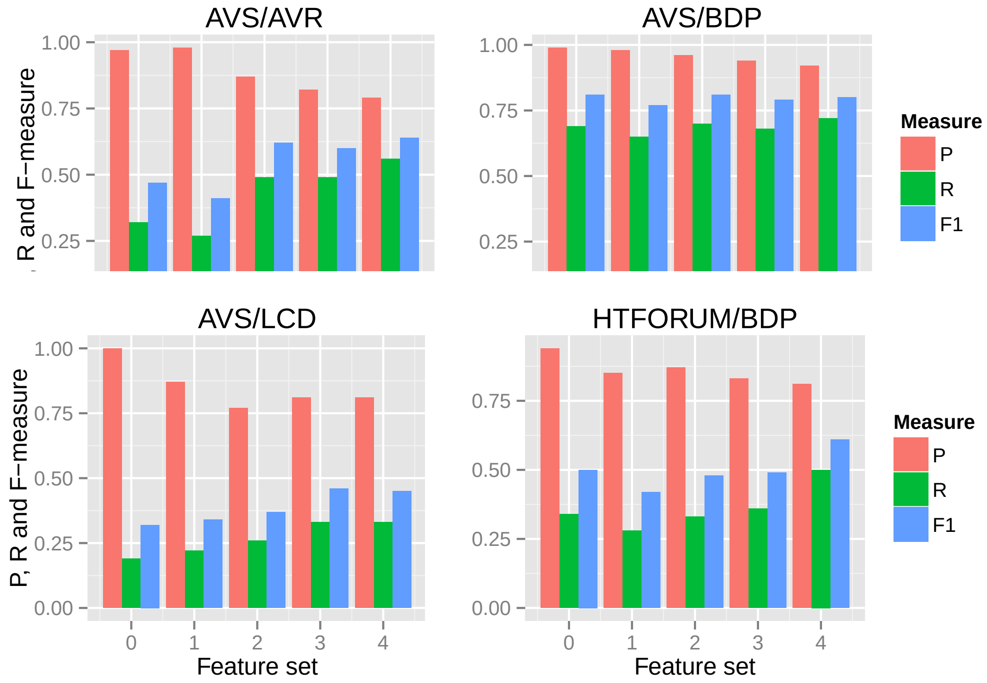


Figure 3.2: Forward selection results for each dataset.

Similarity Threshold

As our method relies on cosine similarity in the SFDetection algorithm, we arbitrarily set the similarity threshold to 0.5, so that no training to set the best threshold is required. As this choice could affect the final performance of the method, we performed experiments to assert the impact of different threshold values for each combination of tested product category and forum. The results of these experiments are presented in Figure 3.3, which shows the threshold variations for each tested product category and forum. In general, higher similarity values yield higher precision levels albeit lower recall levels. Our arbitrary threshold choice is acceptable for a large range of values without much impact from 0 to 0.75 in all datasets. This is explained by the level of label noise introduced in the training data with low similarity threshold values. In such scenarios the resulting classification models are more general despite

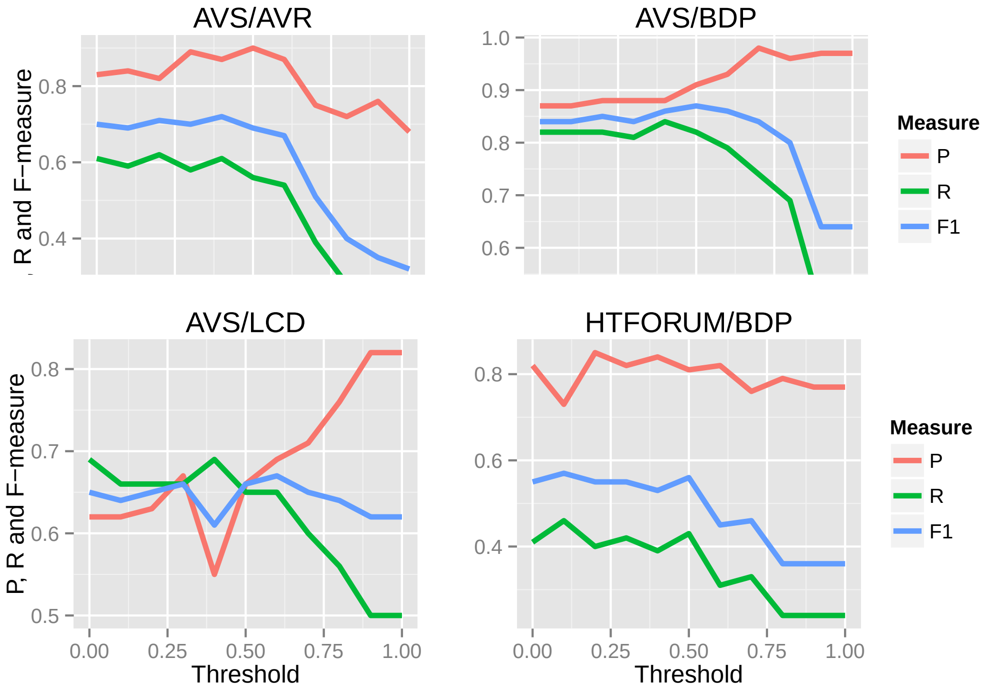


Figure 3.3: Similarity threshold.

the training data noise.

3.5.2 General Results

The first experimental result we report is in Table 3.3, in which ModSpot results are compared with supervised CRF. ModSpot results are from the final CRF generated when our method converges and there is no growth in the seeds set. CRF results are the average from 10-fold cross-validation. The values in bold indicate the highest value achieved for each forum and product category per evaluation metric.

ModSpot achieved higher values for recall and F_1 in all forums and categories compared to CRF. On average, our recall value is approximately 19% higher while the F_1 value is approximately 12% higher. This is a direct result of our Surface Form Detection algorithm, by which significant training exam-

Forum	Category	Method	P	R	F
AVS	AVR	CRF	**0.77**	0.55	0.63
		ModSpot	**0.77**	**0.69**	**0.73**
AVS	BDP	CRF	**0.93**	0.73	0.81
		ModSpot	0.84	**0.88**	**0.86**
AVS	LCD	CRF	**0.81**	0.34	0.47
		ModSpot	0.68	**0.72**	**0.70**
HT	BDP	CRF	0.86	0.55	0.65
		ModSpot	**0.88**	**0.63**	**0.73**

Table 3.3: ModSpot vs. CRF.

Forum	Category	Method	P	R	F
AVS	AVR	ModSpot-SFD	0.67	0.22	0.33
		SFD	0.96	0.19	0.31
AVS	BDP	ModSpot-SFD	0.96	0.25	0.40
		SFD	0.99	0.53	0.69
AVS	LCD	ModSpot-SFD	0.71	0.42	0.53
		SFD	0.95	0.41	0.57
HT	BDP	ModSpot-SFD	0.87	0.51	0.65
		SFD	1.00	0.37	0.54

Table 3.4: ModSpot with no SFD vs. SFD only.

ples, i.e., automatically labeled sentences, are added to the underlying CRF training set. It is worth stressing that the CRF adopted as our baseline was generated in a supervised way, while ModSpot is semi-supervised.

Table 3.4 highlights the importance of the Surface Form Detection algorithm in ModSpot, showing the results obtained when this procedure is not used. This configuration is equivalent to the methods presented in [Teixeira et al., 2011, Putthividhya and Hu, 2011]. These results correspond to lines labeled "ModSpot-SFD". Notice also that in two forum/category pairs ModSpot achieved higher or equal precision in comparison to CRF and ModSpot-SFD, even in spite of a higher recall.

Also, in Table 3.4, to account for the accuracy of the algorithm and the effect of our bootstrap and self-training approach, we report the results of Surface Form Detection alone against the manually labeled golden set. In these results we exclude any form of CRF inference and self-training iteration performed in ModSpot. These results correspond to lines labeled “SFD”. Notice that by using Surface Form Detection only, the recall for all datasets is not high because we do not have all products in the initial seeds. This demonstrates the effectiveness of using a CRF in our self-training approach to achieve higher levels of recall. We argue that improving recall is very important because users are interested in a diverse set of products. Failure to find products can have a negative impact in applications such as opinion mining.

3.5.3 Self-Training Results

In Figure 3.4, we detail the results from Table 3.3 by showing the results of each self-training iteration by forum and product category. The results were obtained as follows. At each iteration, we took the current trained CRF, applied it to the sentences in the golden set and calculated the precision, recall and F_1, considering the golden set of the respective forum/category.

Our method converges at around 9-14 iterations, where there is no growth in the seed set. All the experimented datasets exhibit higher recall and F_1 when the method converges. In the datasets AVS/AVR and HT/BDP the precision remains higher than recall throughout the iterations, and in the datasets AVS/BDP and AVS/LCD recall is higher than precision at around 10 iterations. This is caused by newly discovered seeds that are used to annotate new training sentences.

For the sake of comparison, we show in Figure 3.5 the results of the self-training process without the Surface Form Expansion step. In all datasets, recall is lower than precision as the algorithm did not label any surface form variation that users commonly employ. It is clear that the detection procedure helps the overall measures of the whole process. Our method achieved

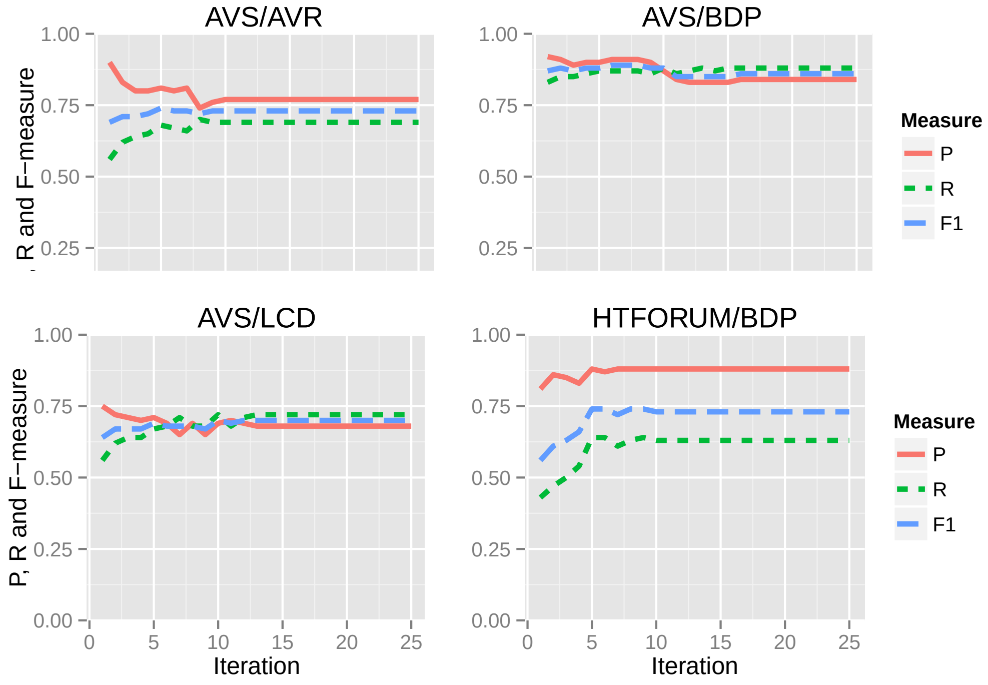


Figure 3.4: Precision, recall and F$_1$ for different datasets per self-training iteration.

higher recall than the simple matching in all datasets because of the detection procedure. This demonstrates the effectiveness of Surface Form Detection.

3.5.4 Seeds

Figure 3.6 shows the number seeds used in each iteration. The first seeds correspond to the initial input seeds manually extracted from products descriptions; further seeds were automatically expanded during the self-training process, and incorporated into our method to annotate new training sentences. The number of seeds is, on average, about 40% larger by the end of the process.

In our experiments, we used all the initial input seeds that were crawled from Amazon.com. We concluded, after an initial experiment with variations of the initial seed set size, that using all the available seeds would yield the best results. This initial experiment varied the size of the initial seed set in

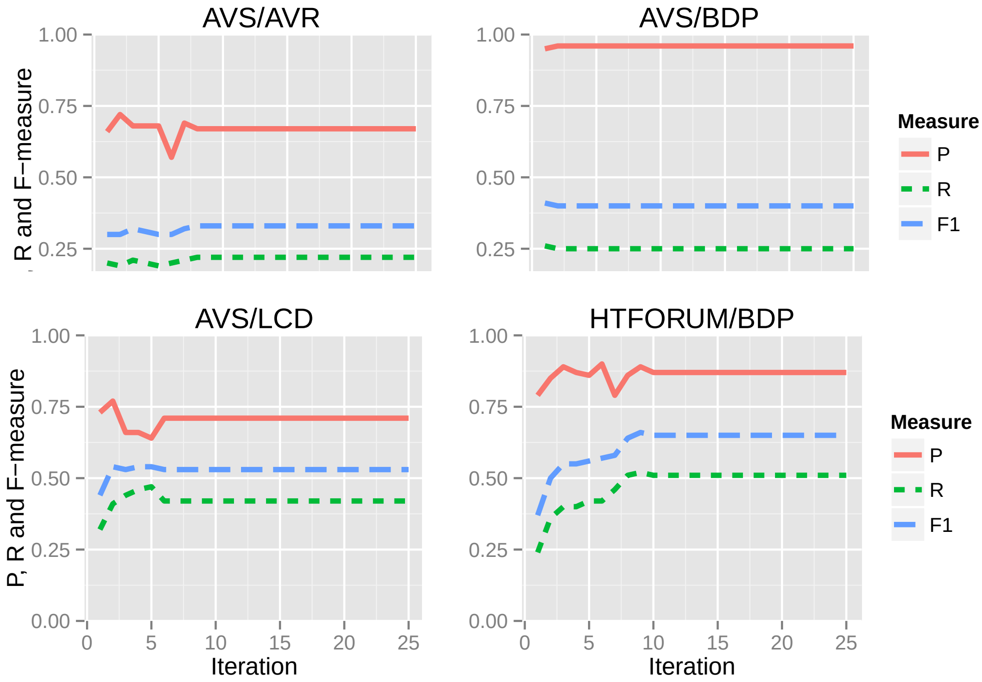


Figure 3.5: Precision, recall and F_1 per self-training iteration without Surface Form Detection.

a 10-fold cross-validation scenario where the seeds were randomly selected for each fold. These results demonstrated that each increment in the initial seed set size yielded better F_1 levels. We stress that obtaining the initial seeds requires very little effort compared to manually labeling a large training set for a CRF.

3.6 Remarks

We presented ModSpot, a method for learning a CRF to undertake the task of identifying model numbers of products. The method is based on a self-training process that requires only a set of initial seed model numbers from consumer products, which means it does not require annotated training sentences to be provided.

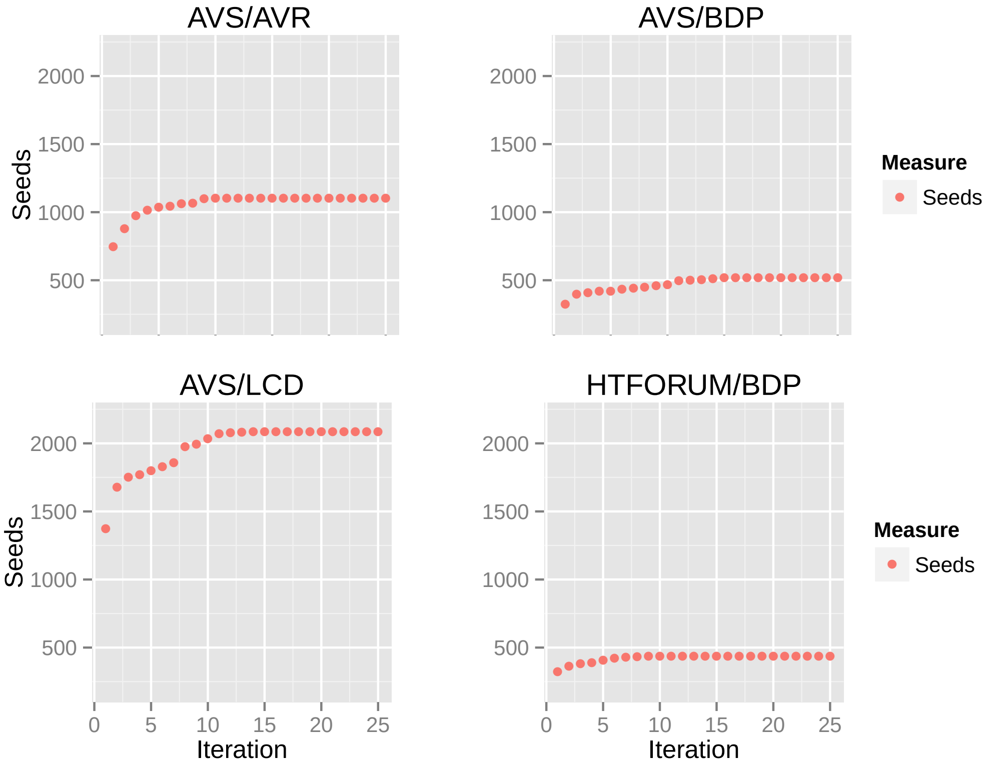


Figure 3.6: Seed growth in each self-training iteration.

Experiments in four settings demonstrated that our method achieved similar or better results when compared to a supervised CRF with the same feature set. All the experimented settings exhibited higher F-measures when our process finished, and the seed set is about 40% larger. In particular, the expansion in seeds performed by the method helped to achieve higher recall levels. In addition, our method converged at around 9-14 iterations, when ModSpot could not identify new seeds.

Although distantly supervised and requiring only a set of seed examples, our first contribution to the sub-task of recognizing product mentions from unstructured textual content was limited to product model numbers. To overcome this limitation, we propose a new method, called *ProdSpot*. We demonstrate that it is possible to build a high-performing product recognition system which labels product mentions from user-generated content by taking as input only unstructured product descriptions, a list of brands, and the target user-generated content. In particular, this is achieved without any manually labeled data by bootstrapping a supervised classifier using a set of examples of product surface forms extracted from the product descriptions. These example surface forms are extracted in an unsupervised manner from product descriptions leveraging only the list of brands. This contribution is presented in Chapter 4.

Chapter 4

Recognizing Product Mentions

In this chapter we present our second contribution toward the goal of product mention recognition. The main task we focus on is recognizing product mentions in user posts. Note that the method presented in this chapter is an evolution of the method presented in Chapter 3.

One of the basic sub-tasks associated with opinion mining is that of *extracting target entities* [Feldman, 2013, Liu, 2012], i.e., entities about which the opinions are made. In general, these entities can be people, brands, companies, places, etc.

In our work, we focus on target entities of a specific, and relevant, type: consumer electronic products, such as smartphones, digital cameras and Bluray players. Such products are the subject of opinions posted by users on a number of posts in discussion forums and retail sites over the Web. More specifically, we address the task of recognizing products of a given category that are mentioned in user reviews and posts.

In our work, we regard the task of identifying product mentions as an instance of the named entity recognition (NER) task [Jakob and Gurevych, 2010, Zhang and Liu, 2011], where the goal is to recognize entity mentions in natural language text. State-of-the-art NER techniques are based on probabilistic graphical models, such as CRF [Sarawagi, 2008]. Although effective, such models are difficult to directly apply to our focus problem, mainly because they require a large amount of labeled data for training. Labeled data is costly to acquire, and this cost is even higher if we consider that there are many distinct user-generated sources and product categories, each one with particularities in terms of lexico-semantic features [Zhang and Liu, 2011]. Also, users often reference the same product using many different surface forms and additional complications arise from the fact that new products (and thus new surface forms) appear very often. Under these conditions, continually providing an adequate volume of representative training instances is an unfeasible task to carry out manually.

We, thus, propose a novel system called *ProdSpot*[1] for learning a named entity extractor to undertake the task of identifying products mentions occurring in user posts. For enabling the learning process, it relies on a set of product descriptions from a consumer products catalog in the same category as the discussion forum (e.g., a smartphone product catalog for a smartphone discussion forum), thus not requiring annotated training sentences from the target forum to be provided. In ProdSpot, the category is implicitly determined by the provided catalog.

In a nutshell, ProdSpot goes through following steps. Initially, typical surface forms used as mentions to products are extracted from a set of product descriptions. Then, given a collection of user posts, the method identifies sentences that contain the extracted surface forms. To improve the quality of

[1] **Prod**uct **Spot**ter.

the extracted mentions, a cluster-based filtering strategy is applied, to detect and filter out possible false examples, which could compromise the precision of the generated model. Finally, to avoid overfitting, our method uses the initial set of sentences to produce more general and diverse set of synthetic sentences. It is this final set of synthetic sentences that will constitute the training set for learning a product mention recognition model.

Experiments in three different settings demonstrate that ProdSpot achieves results similar to those of a supervised CRF model with the same feature set. Our best result for precision, in the smartphone category, is only 9% lower than a supervised CRF model, while our recall level is higher by approximately 7% is two distinct product categories. We also show that these results are directly influenced by our filtering and synthesis strategies, which, respectively, filters errors in bootstrapped samples, and generates new synthetic training examples.

4.1 ProdSpot

In this section, we present an overview of ProdSpot, its main steps, and the techniques we applied in each step. Figure 4.1 illustrates ProdSpot's architecture. Each step of the method is identified by a number used to reference it in the text.

Given a *Forum Text Collection*, that is, a collection of posts from a Web forum on some product category, the ultimate goal is to identify mentions to products made by users in these posts.

ProdSpot is based on a *distant supervision* strategy, and, as it is common in other entity recognition methods based on such an strategy (e.g., [Teixeira et al., 2011,Vlachos and Gasperin, 2006]), it relies on a bootstrapping process for automatically annotating training examples taken from a given input text.

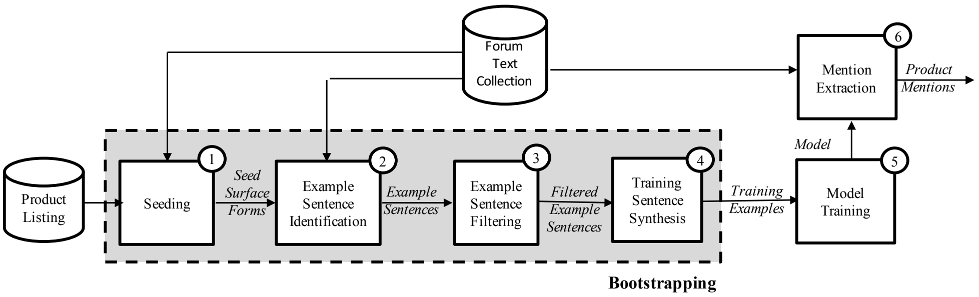


Figure 4.1: The ProdSpot Architecture.

Automatically generating training data is particularly appealing in our case as it replaces costly and manual user labor. In fact, this cost is even higher if one considers that we often deal with many distinct product categories and user-generated sources.

In our method, the bootstrapping process is divided into four distinct steps, corresponding to Steps 1 to 4 in Figure 4.1. They aim at producing diverse and representative training data, and constitute our main contributions.

The process starts with *Seeding* (Step 1). This step relies on a given *Product Listing* containing a list of unstructured *Product Offers*. From such a listing, the method tries to identify examples of product surface forms, which we call *Seed Surface Forms*. These seed surface forms are representations of some typical forms the users employ while mentioning products.

Next, *Example Sentence Identification* (Step 2) scans the Forum Text Collection looking for sentences that can be used as training. It looks for sentences that contain at least one complete Seed Surface Form. This step outputs a set of *Example Sentences*, in which each token that corresponds to a Seed Surface Form is labeled as a product mention example. All other tokens are labeled as non-product mention examples, i.e., other tokens.

As we detail later, although effective in general, Step 2 may generate Examples Sentences with mislabeled tokens. If used for training, these sentences

1. **Apple iPhone 5** Unlocked Smartphone, 16GB, Black
2. **LG Nexus 4** 16GB GSM Unlocked Black
3. **Nokia Lumia 610** 8Gb Black WiFi Windows Unlocked QuadBand 3G Cell Phone
4. **Samsung Galaxy S5**, Black 16GB (Verizon Wireless)

(a)

1. *“I^O have^O a^O **Apple^P iPhone^P 5^P** for^O sale^O.”*
2. *“I^O did^O consider^O an^O **Apple^P iPhone^P 5^P**, because^O they^O were^O about^O to^O be^O released^O after^O the^O **Nexus^O** 4^O.”*
3. *“the^O **samsung^P galaxy^P s5^P** rocks^O at^O taking^O pictures^O!”*

(b)

1. *“I^O have^O a^O **galaxy^P s5^P** for^O sale^O.”*
2. *“the^O **5^P** rocks^O at^O taking^O pictures^O!”*

(c)

Figure 4.2: Example of a simple product listing, sentences identified from the text of the target forum, and a generated sentence.

would compromise the final model. To remedy this situation, ProdSpot includes an *Example Sentence Filtering* (Step 3). This step uses a cluster-based filtering strategy that aims at removing example sentences that include tokens that are likely to be mislabeled. This step outputs a set of *Filtered Example Sentences* that contains only sentences that passed the filtering criteria.

In initial experiments, we detected that the filtered process carried out in this step can eventually yield to some bias on the training sets. To mitigate this problem, we introduce in ProdSpot a *Training Sentence Synthesis* (Step 4), which generates new synthetic sentences by taking advantage of available domain knowledge and the seed surface forms identified during Step 2.

We note that Steps 3 and 4 aim at making our method able to deal with mislabeled examples in the automatically annotated training data. In fact, this is a well-know drawback in previous work ([Teixeira et al., 2011]). The strategies we apply yield a diverse, representative and likely error-free training set in a distantly supervised fashion

In Step 5, we train a supervised classifier using the bootstrapped training data. More specifically, our method uses a conditional random fields (CRF) [Lafferty et al., 2001] classifier. Finally, in Step 6, the resulting classification model can be applied to perform the extraction of product mentions.

Notice that ProdSpot does not handle cases in which product mentions are made using pronouns, since it would require anaphora resolution techniques which we consider out of the scope of our work.

In the next sections, we detail each of these steps.

4.1.1 Seeding

The first step in our method aims at producing a number of *Seed Surface Forms* (Figure 4.1). Our strategy for this consists of identifying examples of surface forms that may occur in a given *Product Listing* composed of unstructured *Product Offers*. An example of such a listing is presented in Figure 4.2(a), which illustrates 4 distinct product offers.

Let O be an offer from the product listing. A *Seed Surface Form* is the largest sequence of tokens $T = t_0 t_1 t_2 \dots t_n$ from O such that, (1) T is a noun phrase, that is, every $t_i \in T$ has a POS tag of either *NN* (noun), *NNS* (plural noun), *NNP* (proper noun), *NNPS* (plural proper noun), or *CD* (cardinal number), and the tokens before and after T do not have such POS tags; (2) t_0 is brand name; (3) T occurs with a frequency $f \geq K$ in the Forum Text Collection (see Step 1 in Figure 4.1).

As an example, in the product listing of Figure 4.2(a), all sequences of tokens in boldface can be regarded as seed surface forms, if their frequency in the Forum Text Collection is equal or above K. In our experiments, K was empirically set 10.

Notice that the definition above assumes that brand names are known to fulfil requirement (2). However, in practice, knowing some brand names is sufficient, since our method only needs to identify a few good seeds. Thus, in our method we rely on set of a few tens of fairly well-know brand names that are supplied in advance as a small dictionary. As shown in our experiments, such a small set of brand names was enough to obtain a high-quality set of seeds.

After the seeds from each product offers from the product listing are identified according to the definition above, we conciliate the set of extracted seeds by removing all duplicates. These resulting seeds are given as input to the next step, Example Sentence Identification.

4.1.2 Example Sentence Identification

Given a set of Seed Surface Forms, this step aims at automatically identifying and annotating sentences to be used as training examples. These sentences are extracted from the Forum Text Collection (Figure 4.1).

A sentence from the Forum Text Collection is taken as a training example if it has at least one substring that match exactly one of the Seed Surface Forms generated in the previous step. All tokens of the matching substring are annotated with a product mention label and all the remaining tokens of the sentence are annotated with a non-product mention label.

In Figure 4.2(b) we illustrate three annotated sentences that result from this step when taking as input the Seed Surface Forms from Figure 4.2(a). Product mentions are marked in bold and the label of each token, product mention or non-product mention (other), is identified in a superscript.

Consider the tokens composing mention **Nexus 4** in the second sentence of Figure 4.2(b), which were annotated with the non-product mention label. Indeed, this mention is incomplete in comparison to the seed surface form in the second line of the listing in Figure 4.2(a). However, these tokens form, in fact, a true product mention. Mislabelling these tokens in an example sentence is likely to harm the training process and the generated extraction model. ProdSpot addresses problems such as this in the Filtering step, discussed next.

4.1.3 Filtering

The previous example shows that the greedy strategy adopted by Step 3 can generate mislabelling. This mislabelling is simply due to fact that the Seeding step cannot account for all the possible surface forms employed by users while mentioning products. The Filtering step (Step 4), aims at detecting and discarding example sentences with mislabeled tokens. This step uses a cluster-based strategy inspired by the well-known nearest centroid (prototype) classifier [Hastie et al., 2009].

Let S be the set of all example sentences identified in the previous step. For each token t from the sentences in S, let $\mathbf{t}$ be a feature vector corresponding to t. Finally, let $\mathbf{p}$ and $\mathbf{n}$ be the *centroids* of all the vectors of tokens that received *product* and *non-product* labels, respectively, in the sentences they occur.

A token t is said to be *wary* if t is labeled as *product* and $d(\mathbf{t}, \mathbf{n}) < d(\mathbf{t}, \mathbf{p})$ or t is labeled as *non-product* and $d(\mathbf{t}, \mathbf{n}) > d(\mathbf{t}, \mathbf{p})$, where d is the Euclidean distance between the two vectors. The filtering process consists of removing from S all example sentences that contain at least one *wary token.*

Intuitively, a token t is considered as wary if it is “closer” to the centroid of the tokens labeled differently than t. We assume that, in such cases, t is likely to be mislabeled.

Intuitively, a token t is considered as wary if it is a negative token whose feature vector $\mathbf{t}$ is "closer" to the vectors of positive tokens, represented by vector $\mathbf{p}$, than to the vectors of negative tokens, represented by vector $\mathbf{n}$. We assume that, in such cases, t is likely to be mislabeled.

Notice that by filtering out any sentence that has even a single wary token, we deliberately assign a conservative character to ProdSpot, that is, sentences that are likely to be mislabeled are discarded. We do so because we want to avoid providing mislabeled sentences to train the extraction model.

In our current implementation, the token vectors (e.g., $\mathbf{t}$,$\mathbf{p}$,$\mathbf{n}$) are built from a set of common features used for information extraction models (e.g, POS tag, brown cluster prefix, words with numbers, words starting with uppercase, etc.).

4.1.4 Synthetic Examples

Another problem we want to avoid in the training data we generate with ProdSpot is the lack of generalization. This problem may occur mainly because example sentences are obtained by searching for seed surface forms that occur in the input forum text collection. Thus, it can be the case that some seed surface forms are highly correlated with certain terms in the sentences. This may induce the classifiers to identify patterns in sentences that are not general enough. Also, seed surface forms represent only one of the many different surface forms or entity expressions that users typically employ while mentioning products.

To avoid this, we apply an strategy to replace the example sentences that result from the previous step with new *Synthetic Examples Sentences.* Let S=$\{s_1, \ldots, s_n\}$ be the set of example sentences resulting from the Filtering step, and let f_{s_i} be the surface form it contains. Also, let $F{=}\{f_{s_1}, \ldots, f_{s_m}\}$ be the set of all seed surface forms from S.

For each sentence $s_i{\in}S$, we generate a synthetic sentence s'_i, in which the surface form f_{s_i} is replaced by f'_{s_j}, where index $1{\leq}j{\leq}m$ is randomly generated and f'_{s_j} results from randomly removing $k{<}|f_{s_j}|$ tokens from $f_{s_j}{\in}F$.

The result is a set of synthetic training sentences $S'{=}\{s'_1, \ldots, s'_n\}$ which is finally supplied as a training set. As verified in experiments we carried out and report here, this strategy for generating synthetic training sentences is indeed very helpful to help the product mention extraction process generalize.

4.2 Experimental Results

We now present an empirical evaluation of ProdSpot on the task of detecting product mentions. We start by reporting the results achieved by our method and a supervised baseline model in an experimental dataset build from real data gathered from the Web. Besides presenting end-to-end results, we also present experiments that show the contribution of each step of the method to reach these results. Finally, we report a experimental comparison we carried out with GREN [Yao and Sun, 2016], a method designed to recognize mobile phone names from Internet forums.

4.2.1 General Results

In this section we report the results achieved by ProdSpot in the task of identifying product mentions in Web forums posts. To better assess the relevance of these results, we compare then with those obtained using a fully supervised CRF model in the same posts.

Experimental Dataset

For this experiment, we manually build a dataset we called the *ProdSpot dataset*. The dataset is a collection of posts related to smartphones (SMRT), digital cameras (CAM), and Blu-ray players (BDP) crawled during a two month period between September and October 2014 from three popular forums on the Web, namely, Howard Forums, Digital Photography Review Forums and AVS Forum, respectively. Howard Forums is an influential Web site that hosts a discussion board dedicated to mobile phones, with over one million members and more than 8 million posts. Digital Photography Review Forums hosts digital photography forums with approximately 40 million posts in 3,6 million threads. AVS Forum is an influential Web site that hosts forums on electronic equipment with over one million members and more than 20 million posts.

From each product category/forum, we sampled 250 posts. Each post was further split into sentences for sequence classification. Sentence split was done at punctuation boundaries. By sampling 250 posts, we were able to achieve a broad coverage of different products in each category. Each sentence was then manually labeled to form our golden set.

The annotation criteria used was labeling as product every token corresponding to a product mention from the corresponding category, even if the token is only a 1-gram corresponding to a brand name that is in fact referring to a product. One example of a mention with a brand-only token is found in the following sentence: “Keeping with the idea that the Nikon has the better kit lens”, where “Nikon” was labelled as a product mention.

Before presenting the results of our experiments, we present an analysis of important characteristics of the ProdSpot dataset regarding the problem we address.

Category	Sentences	Product mentions	Sentences w/ mentions	Avg. mentions per sentence
SMRT	1014	211	156 (15.4%)	1.4
CAM	1447	879	562 (38.8%)	1.6
BDP	1420	545	408 (28.7%)	1.3

Table 4.1: Product mention statistics for each category.

Table 4.1 shows the following statistics for each product category separately: total amount of sentences, amount of product mentions, amount of sentences with at least one product mention and average amount of mentions per sentence.

Posts in the SMRT category have less sentences when compared to the other two categories. This means that user posts in SMRT are shorter, with less text overall. In all posts, there is a large amount of product mentions, although again the SMRT category exhibits less mentions when compared to the CAM and BDP categories. Less product mentions reflects on less sentences with mentions for SMRT, with 15% of the sentences containing at least one product mention; with CAM exhibiting the higher amount of sentences with mentions at 39%. Interestingly, all categories exhibit similar average mentions per sentence, i.e., on average, more than one product is mentioned in a sentence. Average mentions per sentence above 1.0 is due to users mentioning many products in a same sentence, likely mentioning competing or related products.

Further product mention characterization is provided in Figure 4.3. For each dataset we manually inspected each mention and, based on the tokens it contains, grouped the mention into one of the following categories: brand and model, brand only, model only, partial model, and acronym.

Brand and model represents the surface forms where users employ the most

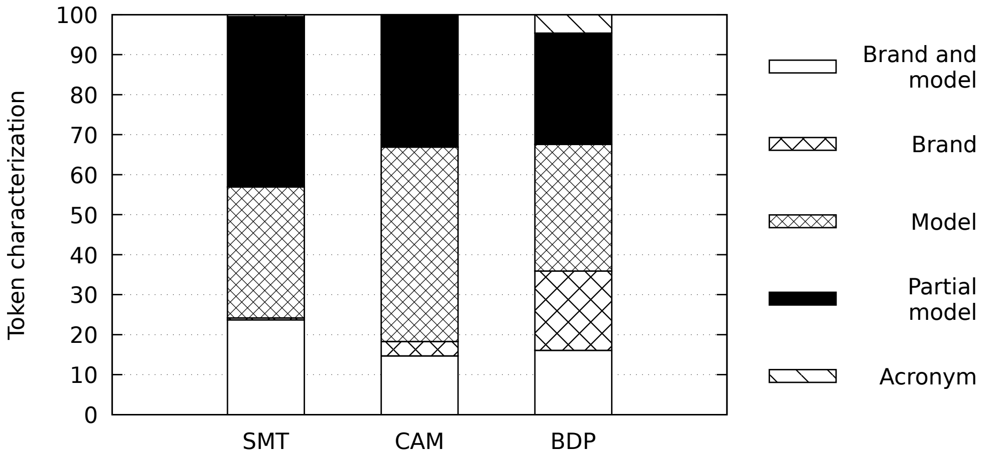


Figure 4.3: Product mention characterization for each category.

detailed product mention by specifying the brand followed by the product model, e.g., “LG Nexus 4”. Brand only mentions are characterized by only containing brand tokens, without direct reference to the product model. Model only are the surface forms that represent mentions where only the product models are used, e.g., “Nexus 4”. Partial model represents mentions where users employ part of the product model. One such example is “4” as referring the *LG Nexus 4* smartphone. Acronym are the mentions where users employ an acronym for the product mention, e.g., “PS3” for “PlayStation 3”.

From the figure we observe that the distribution of mention categories is different for each dataset/category. Brand and model mentions occur more in the SMTR category, at nearly 24%, when compared to CAM (15%) and BDP (16%). Brand only mentions occur much more in BDP that any other category, approximately 19%, while CAM has only 4% of such mentions and SMRT has only one mention. Model and partial model are prominent mention categories in the dataset, and represent combined approximately 60% of the mentions in each category. Acronyms are mostly used in the BDP category, with just 1 use in the other categories.

Category	Product offer
SMRT	Apple iPhone 4 32GB (Black) - Verizon
SMRT	Apple iPhone 4 8GB (Black) - Sprint
SMRT	Samsung Galaxy S6 Edge SM-G925 Factory Unlocked Cellphone, International Version, 32GB, Black
CAM	Nikon COOLPIX P520 18.1 MP CMOS Digital Camera with 42x Zoom Lens and Full HD 1080p Video (Black)
CAM	Olympus Evolt E520 10MP Digital SLR Camera with Image Stabilization w/ 14-42mm f/3.5-5.6 Zuiko Lens
CAM	Canon PowerShot ELPH 520 HS 10.1 MP CMOS Digital Camera with 12x Optical Image Stabilized Zoom 28mm
BDP	Sony BDP-S550 1080p Blu-ray Player (2008 Model)
BDP	LG Electronics BP550 Blu-Ray Player
BDP	Sony BDP-S5500 3D Streaming Blu-Ray Disc Player with TRI-LUMINOS Technology

Table 4.2: Examples of product offers in the ProdSpot dataset.

The ProdSpot dataset also includes products listings collected from *Amazon.com* to extract seed surface forms. Examples of product descriptions from this listings are presented in Table 4.2. For each category, we crawled its product listing pages and extracted product descriptions. For SMRT we crawled 1082 descriptions, while CAM has a total of 1587 descriptions, and BDP has 521 descriptions. Descriptions where not deduplicated and, as such, there may exist more than one description of the same product.

Evaluation Metrics

To evaluate our method, we used the well-known *precision*, *recall*, and F_1 metrics as defined in Section 3.5.1.

CRF Configuration

The CRF model generated by ProdSpot and the fully supervised one used as a baseline use the same set of features and configuration, learned for each category. The features we adopted are widely used in previous work [Zhang and

Set	Description
0	Current token
1	Tokens in a context window of size 3
2	Part-of-speech tag of the current token and of the tokens in the context window
3	Token begins with uppercase, token is all uppercase and token has a character that is uppercase
4	Token is numeric, token is a combination of alphanumeric characters and token has punctuation
5	Brown cluster prefixes

Table 4.3: Features used by the CRF models (supervised and distantly supervised).

Liu, 2011,Jakob and Gurevych, 2010,Sarawagi, 2008]. Although CRF models are flexible enough to allow specific features for different domains, we used the same set of features and configurations in all experiments. These features are described in Table 4.3 and correspond to a setup similar to [Ratinov and Roth, 2009], including Brown cluster prefixes. We used the CRF implementation presented in [Lavergne et al., 2010], trained with stochastic gradient descent and L1 regularization.

The Brown algorithm is an unsupervised method that generates word clusters from unlabeled text [Brown et al., 1992]. These word clusters are hierarchical, producing a binary tree from word contexts as they appear in the unlabeled text. For example, since the words “Galaxy” and “BlackBerry” appear in similar contexts, the Brown algorithm will assign them to the same cluster. Successful abstraction of both as products related tokens, addresses the data sparsity problem common in natural language processing tasks. Within the binary tree produced by the algorithm, each word can be uniquely identified by its path from the root. This path is represented by a string of 0s and 1s. Paths of different depths along the path from the root to the word provide different levels of word abstraction. For example, paths at depth 4 closely correspond to part-of-speech (POS) tags. In this work, as in [Ratinov and

Category	Method	P	R	F_1
SMRT	CRF	**0.93**	0.80	**0.86**
	ProdSpot	0.84	**0.87**	**0.86**
CAM	CRF	**0.96**	**0.89**	**0.92**
	ProdSpot	0.83	0.83	0.83
BDP	CRF	**0.87**	0.90	0.87
	ProdSpot	0.82	**0.96**	**0.88**

Table 4.4: ProdSpot vs. CRF.

Roth, 2009], we used path prefixes of length 4, 6, 10, and 20.

Results

These results for this first experiment are presented in Table 4.4, and are the averages of a 10-fold cross-validation. The values in bold indicate the highest value achieved for each forum/product category per evaluation metric. ProdSpot results were achieved by learning a CRF model with the output training data after the Bootstrapping step.

ProdSpot achieved competitive values in all categories when compared to a supervised CRF model, despite being distantly supervised and not requiring any user input. Our best result for precision, in the SMRT category, is only 9% lower than the CRF model, while our recall level is higher by approximately 7% is two product categories. The F_1 result for the SMRT category is the same as the supervised model result, with our method achieving higher recall albeit lower precision in this category.

These results are directly influenced by our Filtering and Training Sentence Synthesis steps, which, respectively, filters errors in bootstrapped samples, and generates new synthetic training examples. The Synthesis step takes advantage of available domain knowledge and the seed surface forms previously identified in the corpus to change data in the underlying CRF training set, thus enabling better model generalization. In the following sections we present the results after each Bootstrapping step. It is worth stressing that the CRF model adopted as our baseline was generated in a supervised way, while ProdSpot is distantly supervised.

We now examine in detail the mislabeling errors, in each category, that affect the precision results. In the SMRT category, of the 52 mislabeled (false-positive) tokens, 25% are brand name, and 14% are products from other categories not directly related to the target forum. One interesting observation from this category is that product model names use more common words, such as “note”, “dash” and ”bold“, when compared to the other categories (CAM and BDP). These classification mistakes represent 31% of the overall errors. In the CAM category, of the 144 mislabeled (false-negative) tokens, 83% are brand names, and 10% are products from other categories not directly related to the target forum such as lens.

Finally, in the BDP category, of the 48 mislabeled (false-positive) tokens, 63% are brand names, and 17% are products from other categories not directly related to the target forum such as audio/video receivers and televisions. In this category, brand name tokens are particularly ambiguous, as these tokens may mention a product or the brand itself. As seen in the mention characterization presented in Figure 4.3, 20% of the mentions are are characterized by only containing brand tokens, without direct reference to the product model.

These misclassification errors (false-positive) are likely due to the lack of brand tokens as negative examples, thus the resulting models are biased into classifying brand tokens as products while they may actually represent brand mentions.

4.2.2 Detailed Results

In this section we present an empirical evaluation of each Bootstrapping step in ProdSpot. For each category, we provide the relevant output results related to each step. For example, we present the amount of extracted seeds for the Seeding step. Also, we present results for mention extraction from a CRF model trained with the output of each step, except for the Seeding step. This aims at showing the contribution of each Bootstrapping step towards the final goal of generating training data for learning the extraction model in a distantly supervised way. The Seeding step is evaluated by directly labeling the test set with the seeds that were identified from product offers.

Seeding

The first step in our method aims at producing a number of seed surface forms that may occur in a given product listing composed of unstructured product offers. As such, we present in Table 4.5 the amount of unique seed surface forms produced by the step. As we can see, the Seeding step produced a significative amount of several tens unique seed surface forms for each category.

Table 4.5 also shows the precision (P), recall (R) and F_1 results obtained after labelling the test set using only the seeds identified from each category offers. We observe that the seed surface forms produced by the Seeding step achieved very high precision values in all categories while exhibiting very low recall levels. All categories exhibit perfect precision. On average, the recall level is at 2.7. For instance, in the BDP category only 3 seed surface forms were found, the lowest recall level for all categories.

These results indicate that the Seeding step produces high-quality seed surface forms. High-quality in our scenario means seeds without false-negative tokens. The low recall levels are explained by our requirement that a seed

Category	Offers	Seeds	P	R	F_1
SMRT	1082	94	1.00	0.04	0.08
CAM	1587	56	1.00	0.03	0.06
BDP	521	24	1.00	0.01	0.01

Table 4.5: Results by seed labelling.

Category	Sentences	P	R	F_1
SMRT	8049	1.00	0.18	0.30
CAM	2433	1.00	0.08	0.14
BDP	1238	1.00	0.03	0.06

Table 4.6: Classifier results after bootstrapping

surface form starts with a brand name. Following the mention characterization used in Figure 4.3, the seeds identified in the Seeding step are characterized by being composed of brand and model. As that figure indicates, mentions of this this kind are one of the least frequent forms users typically mention products.

Note from Table 4.5 that the amount of seeds for each category is proportional to its recall level. Higher amounts of seeds yield higher recall levels.

Example Sentence Identification

Given the set of seed surface forms, the Example Sentence Identification step aims at automatically identifying and annotating sentences to be used as training examples for the CRF model. As such, we present in Table 4.6 the amount of training sentences produced by this step. As it can be seen, thousands training sentences were identified for each category.

Table 4.6 also shows the precision (P), recall (R), and F_1 values that would be achieved by the CRF model trained with these sentences. As expected, we observe that the classification result achieved very high precision values in

Set	Description
1	Tokens in a context window of size 3
2	Part-of-speech tag of the tokens in the context window
3	Brown cluster prefixes

Table 4.7: Features used by the Filtering step.

all categories while exhibiting low recall levels. All categories exbibit 1.0 of precision. This classification result shows that the model is very biased, and unable to generalize. This is a result of our greedy strategy that, by identifying examples sentences from forum text, generate false negatives. As explained earlier, this mislabelling is simply due to fact that the Seeding step cannot account for all the possible surface forms employed by users while mentioning products.

Note from Table 4.6 that the level of recall for each category is proportional to the amount of training sentences produced. This indicates that the CRF model is able to generalize better with more training sentences.

Filtering

The Filtering step, aims at detecting mislabeled example sentences. As such, we present the amount of training sentences that remain in the training set after the step. After filtering 4570 sentences remain for the SMRT category, 683 sentences for CAM, and 594 sentences for the BDP category. These final amounts of training sentences represent, respectively for each category, 57%, 28%, and 48% of the initial training size after the Example Sentences Identification step.

To filter mislabeled training sentences, we use a simplified set of features compared to our CRF model. These features are described in Table 4.7.

Table 4.8 shows the results obtained after the Filtering step. For the sake

Category	Step	P	R	F_1
SMRT	Filtering	**0.87**	0.82	0.84
	Synthesis	0.84	**0.87**	**0.86**
CAM	Filtering	0.78	0.59	0.67
	Synthesis	**0.83**	**0.83**	**0.83**
BDP	Filtering	**0.86**	0.72	0.78
	Synthesis	0.82	**0.96**	**0.88**

Table 4.8: Classifier result after filtering and synthesis.

of comparison, we include the results achieved by learning a CRF model with the output training data after the Synthesis step, i.e., the final ProdSpot generated model. We observe that the classification result achieved represent much better results when compared to the non-filtered results. The best result for precision, in the SMRT category, is 4% higher than the final ProdSpot model, while the recall level is lower by approximately 6%. The F_1 results for all categories are lower than the results for the final CRF models trained with filtered and synthetic examples. This is mostly related to recall scores being lower than the final CRF models results, as the training examples are still biased with mention examples only containing brand and model.

4.2.3 Comparison to GREN

In this section, we report the results obtained by ProdSpot in comparison to those obtained by GREN [Yao and Sun, 2016]. For this experiment, we use the same dataset used to evaluate GREN, called here the *GREN dataset*, which was graciously provided by its authors.

GREN is a method designed to recognize mobile phone names from Internet forums, much like the task performed by ProdSpot in the SMRT category. We consider GREN to be a suitable baseline since its objective is the same as in ProdSpot, although limited only to the smartphone category, and a limited selection of products.

To form this dataset, the authors crawled posts from the HardwareZone Forums related only to smartphones. HardwareZone is a popular Web forum from Singapore with about 700,000 users. For each smartphone in a selection of 20 smartphones, the authors sampled a forum discussion thread with about 100 posts. The posts were further split into sentences that were also manually labeled to form the golden set.

Differently than our annotation criteria, the authors disregard as product mention the tokens that do not refer to any known smartphone model. In other words, only mentions to the 20 smartphones in the selection were considered as product mentions. One example of a mention ignored by the authors in their evaluation is found in the following sentence: "Buy now, there will be new iphone6 when your next contract end.", where "iphone6" is clearly a product mention, but ignored during evaluation. For the sake of comparison, we report our results in the dataset using the same criteria as GREN's authors.

Table 4.9 shows the results obtained by our method, two methods proposed by the authors (GREN and GREN-NC), and a supervised CRF model cross-validated using the GREN dataset. The supervised CRF model uses the same set of features and configuration as the model generated by ProdSpot. The configuration is also the same as the one used in the previous experiment. The model generated by ProdSpot was learned using the same examples from the previous experiment for the SMRT category. We did not have access to the whole forum text from the GREN dataset to enable the whole ProdSpot bootstrapping process.

GREN generates candidate product mentions in forum text that are later classified as indeed referring to a product. Classification is done on modified

Method	P	R	F_1
ProdSpot	0.76	0.86	0.81
GREN	0.82	**0.87**	0.85
GREN-NC	0.93	0.52	0.67
CRF	**0.97**	0.78	**0.86**

Table 4.9: Classification result using the GREN dataset

sentences extracted from text where each candidate (sequence of tokens) is rewritten as a single token.

We stress that the GREN classification model is learned in a supervised way, where a set of manually annotated negative examples are used to select training examples. GREN-NC, is similar to GREN, however, sentences are kept in their original form as no candidate mention is generated.

We can see that ProdSpot achieved competitive result compared to GREN and GREN-NC, despite being distantly supervised and not requiring any user input. The F_1 result is only 4% lower than GREN while being 20% higher than GREN-NC. Of the 482 mislabeled (false-positive) tokens, 57% are brand names. As expected, the highest precision was achieved by the supervised CRF model.

4.3 Remarks

In this work, we presented a novel method, called ProdSpot, to undertake one of the basic sub-tasks associated with opinion mining: extraction of target entities, i.e., entities about which the opinions are made. We focused on target entities of a specific and relevant type: consumer electronic products, namely smartphones, digital cameras and Blu-ray players. Such products are the main subject of opinions posted by users on a number of posts in discussion forums and retail sites on the Web. More specifically, we addressed the task of recognizing products of a given category that are mentioned in user reviews and posts.

Experiments were executed to compare our approach to state-of-the-art solutions: a supervised CRF model, GREN and GREN-NC. GREN and GREN-NC are methods designed to recognize mobile phone names from Internet forums.

ProdSpot achieved competitive values in all categories when compared to a supervised CRF model, despite being distantly supervised and not requiring any user input. Our best result for precision, in the smartphone category, is only 9% lower than the CRF model, while our recall level is higher by approximately 7% is two product categories. These results are directly influenced by our Filtering and Training Sentence Synthesis steps, which, respectively, filter errors in bootstrapped samples, and generate new synthetic training examples. On average, 57% of classification errors are attributed to tokens which are brand names.

When compared to GREN and GREN-NC, despite being distantly supervised and not requiring any user input, ProdSpot also achieved competitive results. The F_1 result was only 4% lower than GREN while being 20% higher than GREN-NC. Again, of the mislabeled (false-positive) tokens, 57% are brand names.

Once product mentions have been recognized, the next step is to link these mentions to their corresponding products from a catalog. We argue that this problem can be effectively solved using a set of evidences that can be extracted from the social media content and product descriptions. More specifically, we show which features should be used, how they can be extracted, and then how to combine them through machine learning techniques. This contribution is presented in Chapter 5.

Chapter 5

Linking Product Mentions to Products on a Catalog

In this chapter we present our contribution to the second sub-task of linking recognized mentions to their real world counterpart. The main task we focus on is linking products which are mentioned in user posts to the corresponding product in a catalog.

We present a method to link product mentions to their respective real-world products. We argue that this problem can be effectively solved using a set of evidences that can be extracted from the social media content and product descriptions. Specifically, we show which features should be used, how they can be extracted, and then how to combine them through machine learning techniques. Our method was applied in a product linking system, called *ProdLink*[1]. ProdLink is an end-to-end solution for product linking, capable of both recognizing product mentions in natural language text from public forum posts and of linking those mentions the entries in a catalog. In this paper, however, we will focus on the actual linking task, and use a

[1] **Prod**uct **Link**er.

standard state-of-the-art solution to recognize the mentions.

Experiments with two different datasets, demonstrate that ProdLink, with its feature set geared toward product mention disambiguation, achieves higher values for precision, recall and F_1 in several product categories, compared to two state-of-the-art baselines. When compared to our best performing baseline, our gains in precision, recall, and F_1 values are approximately 0.17, 0.08, and 0.13, respectively. In particular, we show that contextual information is fundamental to achieve such high levels of precision. All experiments were performed on two distinct datasets, one of which was created by the authors and made available to the community.

Our main contributions are, thus (a) a novel method for the problem of product linking, derived from a thorough analysis of the problem and an exploration of the information that can be used to solve it; (b) a description of the set of features that should be used to disambiguate product mentions in user-generated content; (c) a characterization of how users typically mention products; and (d) a new dataset for product linking research. To the best of our knowledge, no other work as performed such a detailed analysis of the information available in user-generated content for the purpose of product linking. The fact that we were able to reach consistent conclusions using two different datasets, containing several different product categories, confirms that our methodology, i.e. the set of selected features and the classifier used, is robust enough to be used in future product linking tasks.

5.1 Linking Mentions to Products

As it is common in the product linking task, our proposed method will work by (1) discovering product mentions inside a given input text and (2) linking those mentions to actual products in a reference catalog. We will apply supervised

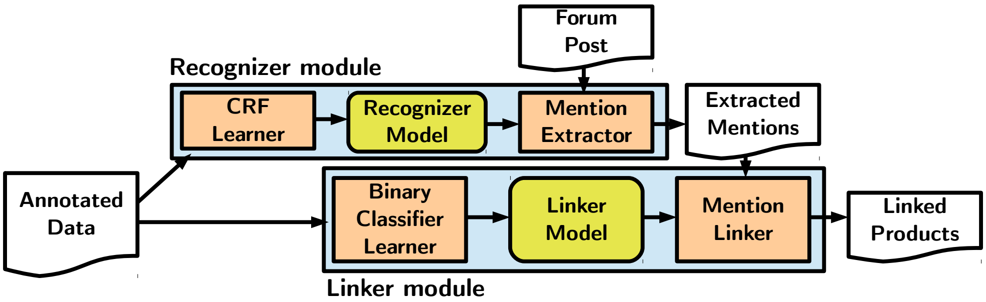


Figure 5.1: The ProdLink Architecture.

machine learning techniques to both tasks, thus fully exploiting the available annotated data and allowing for a more flexible product linking solution. Our proposed method has been implemented into a product linking system, called *ProdLink*. In the following, we will use ProdLink to illustrate our proposal and explain in detail how each of these tasks is performed.

5.1.1 The ProdLink Architecture

Figure 5.1 shows an overview of the ProdLink architecture. ProdLink is composed of two main modules, corresponding to its two main tasks of recognizing product mentions (described in Section 5.1.2) and linking product mentions to entries in a catalog (described in Section 5.1.3). There are also two main aspects to its workflow: building the recognition and linking models and linking the product mentions.

For the training workflow, since we are using a supervised machine learning approach, ProdLink assumes that a set of annotated data is available. This data should contain textual entries, such as forum posts, where each mention to a product is correctly linked to one or more entries in a product catalog. The same data can be used by both the Recognizer module and the Linker module to train their respective models. In the first case, the Recognizer module will extract features from the text of the posts containing each annotated entry, to train a conditional random field classifier. In the latter case, features are extracted from each ⟨product mention,catalog entry⟩ pair to learn a binary classification model, which will yield, as output, a *positive* or *negative* decision on whether the mention corresponds to the catalog entry.

In the linking workflow, once both models are learned, ProdLink takes as input a forum post (or a *thread* of forum posts) and uses the Recognizer module to extract product mentions. The extracted mentions are then given to the Linker module, which compares the mentions to a set of *candidate entries*, taken from the catalog. Each of the ⟨product mention,catalog entry⟩ pairs thus obtained is classified as either a true match or a false match. Candidate entries are selected from the catalog by choosing those that contain the same tokens contained in the mention.

In the following sections we explain in more detail how the models are trained and how mention extraction and classification are processed.

5.1.2 Recognizing Product Mentions

Recognizing product mentions can be achieved with the direct application of Name Entity Recognition techniques [Nadeau and Sekine, 2007]. As in previous work (e.g., [Wu et al., 2012]), here we adopted a conditional random field (CRF) sequence classifier [Sutton and McCallum, 2012]. CRFs are probabilistic graphical models used to predict a sequence of labels for an input sequence of tokens. To achieve this, they consider features derived both from the token itself and from the sequencing and ordering of the labels assigned to neighbor tokens. In our case, a CRF model is trained using sentences from forum posts, which are annotated with product mentions. Afterwards, given a sentence from an unseen post, the trained CRF model must be able to correctly label any product mention that may occur. The training process is performed through stochastic gradient descent with L1 regularization.

Since the focus of our work is on the Linker module of ProdLink, for the Recognizer module we opted to use a standard set of features, known to perform well in other entity recognition tasks. Thus, the set of features used in our mention recognition model are:

Current Token: The current token in the sentence.

Context Tokens: Tokens in a context window of size 6, centered on the current token.

POS Tag: Part-of-speech tag of the current token and of the tokens in the context window.

Token Case: Token begins with uppercase, token is all uppercase, or token has a character that is uppercase.

Token Type: Token is numeric, token is a combination of alphanumeric characters, or token has punctuation.

Once product mentions are discovered, the actual product linking step of our approach can be performed.

5.1.3 Linking Product Mentions

As explained, product linking is performed through a binary classifier. The classifier is used to decide if a given mention in a post refers to a specific product or not. In this scenario, training and testing instances are characterized by pairs $\langle m, p \rangle$, where m is a mention found in a post and p is an entry on a product catalog C. Pairs where mention m actually corresponds to product p are defined as positive, while the remaining pairs are negative. Using this information, a binary classification model is learned for each product category.

Product linking is performed by presenting a pair $\langle m, p\rangle$ to the corresponding classifier, which returns positive if the mention corresponds to the product, or negative otherwise. All $\langle m, p\rangle$ pairs classified as positive are considered as linked. Thus mention m may potentially link to different products from the catalog.

One might wonder if the scheme used to build the training dataset has a class imbalance problem. The class imbalance problem typically occurs when there are much more instances of some class than the other class [He and Garcia, 2008]. In such cases, classifiers tend to be biased into making predictions towards the class with the most number of instances, in our case, the negative class. However, such problem has not been observed in our experiments, where the recall levels achieved by ProdLink are high. If the class imbalance indeed pose a problem for ProdLink the recall levels should be low as there are more negative examples then positive examples.

In our solution, each pair $\langle m, p\rangle$ is defined by a set of statistical features extracted from the textual representation of both m and p. We start, therefore, by explaining how this representation is constructed.

Let t be an unstructured text containing a product mention m and let d be the description of product p from the product catalog C. We represent each t and d as a sequence $x_1, x_2, \ldots, x_n$, where each x_i is a token composed of only letters or only digits. Thus, t and d are represented as sequences of tokens. For example, a product description such as "*LG Electronics BP550 Blu-Ray Player*" would be represented by the sequence of tokens "LG", "Electronics", "BP", "550", "Blu-ray", and "Player".

The features described in the following section are derived from the token sequence for the text containing m and the token sequence for the product description p.

5.1.4 Features for Product Linking

The basic user generated content from forums are *posts*, which are grouped into *threads*. A *thread starter* post is the first post of a thread, and assigns it a *thread title*. *Follow up posts* consist of replies to existing posts. Formally, we model a thread h as a sequence $c_1, c_2, \ldots, c_n$, where each c_i is a post containing unstructured text. Thus, a thread consists of a chronologically ordered list of related posts.

Our supervised learning method relies on a set of 18 features which exploit the characteristics of product descriptions and forums to disambiguate mentions. We now describe each feature and the rationale behind it. To make our feature descriptions easier to follow, we organize the features into groups, according to the type of information they convey.

Description Similarity Features

Description similarity features capture the textual similarity between a product mention m and the description d of a product in the catalog. They are defined as follows.

Exact Match: Takes the value 1 if mention m is an exact match with d, zero otherwise.

Substring Match: Takes the value 1 if mention m is a substring of d, zero otherwise.

Token Subset: Takes the value 1 if the tokens of m are a subset of the tokens of d, zero otherwise.

Token Count: Represents the amount of tokens common to m and d.

Term Subset: Takes the value 1 if the terms of m are a subset of the tokens of d, zero otherwise. We define a *term* as a sequence of alphanumeric characters.

Term Count: Represents the amount of terms common to m and d.

Alpha: Takes the value 1 if m is composed only by alphabetic characters, zero otherwise.

Numeric: Takes the value 1 if m is composed only by digits, zero otherwise.

Positional Features

We observe that typically a product mention is found at the beginning of a product description. Thus, it is important to model explicitly this characteristic as it is possible that a product mention match a string unrelated to the actual product name and model towards the end of the product description. Positional features capture where, in the product description d, the tokens that compose the product mention m occur. They are defined as:

Mention Size Number of tokens that compose the mention. A correct product mention should not be too long nor too short.

First mention token position Position of the first token of the product mention in the product description. We expect the actual product name and model to appear early in the product description.

Last mention token position Position of the last token of the product mention in the product description. This feature should complement the *first mention token position* feature.

Token position distance Distance, in tokens, from the first mention token position to the last mention token position. We expect the tokens that compose the mention to be close in the product description.

Acronym Similarity Features

Users typically employ acronyms to refer to a particular product. For example, the *Samsung Galaxy S3* phone is usually referred to as *SGS3*. Thus, it is important to deal explicitly with this type of surface forms. The following features capture the similarity between a product mention m and the acronym of the product. An acronym for a product p is built by taking the first character from the tokens of its description d.

Acronym Match: Takes the value 1 if mention m is an exact match with the acronym, zero otherwise.

Acronym Subset: Takes the value 1 if the tokens of m are a subset of the tokens of the acronym of p, zero otherwise.

Position of Acronym: Position of the first character of m in the product acronym.

Context Similarity Features

These features measure the similarity between the product description d and the unstructured text surrounding each mention m. We call this surrounding text the context (c) of m. Depending on how we precisely define c, we obtain different contextual features. In this case, our features are:

Post Context Similarity: For this feature, we consider c as the terms occurring in the post where mention m appears.

Thread Title Similarity: For this feature, we consider c as the terms occurring in the title of the thread containing the post where mention m appears.

Thread Similarity: For this feature, we consider c as the terms occurring in all the thread posts previous to the post containing mention m.

All of the context similarity features take the value of the *cosine similarity* between c and d [Baeza-Yates and Ribeiro-Neto, 2011]. More specifically, let $\mathbf{d} = \langle w_{d1}, \cdots, w_{dn} \rangle$ be the vector representation of d and let $\mathbf{c} = \langle w_{c1}, \cdots, c_{cn} \rangle$ be the vector representation of c, where each w_{di} and w_{ci} represent the frequency of token x_i in d and in c, respectively. We define the similarity between d and c as:

$$sim(d, c) = \frac{\sum_{i=1}^{n} w_{di} w_{ci}}{\sqrt{\sum_{i=1}^{n} w_{di}^2} \sqrt{\sum_{i=1}^{n} w_{ci}^2}} \tag{5.1}$$

where n is the size of a common vocabulary in d and c.

We note that the thread similarity features only consider the posts previous to the post containing mention m, since in a real user forum, any liking method would not have access to posts occurring after the current one.

5.2 Experimental Results

We now present an empirical evaluation of ProdLink on the task of linking product mentions to product entries in a catalog. We start by describing the datasets, evaluation metrics and the baseline adopted, including a detailed dataset characterization. Finally we report the results achieved by our method and the baselines.

5.2.1 Setup

We start by reporting the experimental datasets used throughout the experiments, the evaluation metrics and the baseline used.

Experimental Datasets

To evaluate our proposed method, we used two datasets: the *CPROD1 dataset* [Melli and Romming, 2012] and a manually built dataset, containing 50 different products in three different product categories, which we will call the *ProdLink dataset*.

The CPROD1 dataset[2] was first used in the CPROD1 contest, held within ICDM'12. It contains 2111 text items extracted from Web pages or discussion forums, and a list of $15,367,328$ products out of which only 4252 are actually linked to at least one mention. It should be noted that, although this list contains duplicate items, in our experiments, we used it as is and did not perform any pre-processing. Also, since CPROD1 does not provide post threads, in all experiments with this dataset the *Thread Title Similarity* and *Thread Similarity* features were not used.

In our experiments, we did not use the provided split into training and test sets, instead using all the data to perform 5-fold cross-validation. The split into training and testing sets was done by splitting the set of forum posts, i.e. using 4/5 of the posts, together with products therein mentioned, for training and 1/5 of the posts for testing.

The ProdLink dataset[3] contains forum posts commenting on three different product categories of consumer electronics: blu-ray players (BDP), digital cameras (CAM) and smartphones (SMRT). Each category contains a collec-

[2] Available at `https://www.kaggle.com/c/cprod1/data`

[3] Available at `http://shine.icomp.ufam.edu.br/prodlink`

Category	Product Description
BDP	Sony BDP-S550 1080p Blu-ray Player (2008 Model)
BDP	LG Electronics BP550 Blu-Ray Player
BDP	Sony BDP-S5500 3D Streaming Blu-Ray Disc Player with TRI-LUMINOS Technology
CAM	Nikon COOLPIX P520 18.1 MP CMOS Digital Camera with 42x Zoom Lens and Full HD 1080p Video (Black)
CAM	Olympus Evolt E520 10MP Digital SLR Camera with Image Stabilization w/ 14-42mm f/3.5-5.6 Zuiko Lens
CAM	Canon PowerShot ELPH 520 HS 10.1 MP CMOS Digital Camera with 12x Optical Image Stabilized Zoom 28mm
SMRT	HTC Desire 610 8GB Unlocked GSM 4G LTE Quad-Core Android Smartphone
SMRT	Nokia Lumia 610 8Gb Black WiFi Windows Unlocked QuadBand 3G Cell Phone
SMRT	HTC 8x c620E 16GB Unlocked GSM Smartphone - No Warranty - Black - GSM: 850/900/1800/1900 MHz

Table 5.1: Examples of product catalog entries in the ProdLink Dataset.

tion of posts crawled during a two month period between September and October 2014 from three popular forums on the Web, namely, AVS Forum, Digital Photography Review Forums and Howard Forums, respectively. AVS Forum is an influential Web site that hosts forums on electronic equipment with over one million members and more than 20 million posts. Digital Photography Review Forums hosts digital photography forums with approximately 40 million posts in 3,6 million threads. Howard Forums is another influential Web site that hosts a discussion board dedicated to mobile phones, with over one million members and more than 8 million posts.

From each product category and forum, we sampled 250 posts. By doing so, we were able to achieve a broad coverage of different products in each category. The dataset was then manually labeled to form our golden set. For each category, we used 50 products whose catalog descriptions were collected from *Amazon.com*. Examples of these descriptions are presented in Table 5.1. Clearly, one can notice that relying only on string or character matching is likely to fail, as mentions to different products are made using a same surface form. This occurs specially in shorter and more ambiguous mentions.

Evaluation Metrics

To evaluate our method, we used the well-known *precision*, *recall*, and F_1 metrics. Precision is the ratio of correct product links among all predicted links. Recall is the ratio of correct product links among all manually labeled product links. F_1 is the harmonic mean of precision and recall.

More formally, let G be the golden set with manually labeled links and S the result set yielded by the ProdLink system. We define precision (P), recall (R) and F_1 as:

$$P = \frac{|G \cap S|}{|S|} \qquad R = \frac{|G \cap S|}{|G|} \qquad F_1 = 2 \times \frac{(P \times R)}{(P + R)} \tag{5.2}$$

Baseline Methods

To serve as baselines for comparison, we use the methods proposed in [Wu et al., 2012] (Wu) and [Yao and Sun, 2016] (GREN), since the problem they address is quite similar to ours. We implemented the methods according to the descriptions in the respective papers. In the case of GREN, as the method was originally proposed for dealing with cell phones, and required a structured representation of the catalog entries, some adaptations were made to allow experimentation in other product categories. More specifically, since, unlike GREN, our catalog entries are not in a structured format, we do not use the parts of the algorithm that depend on such structure, such as the co-occurrence confidence computation, i.e. we use only the first step of its rule-based name normalization for linking the mentions to the products. We recognize that this does not allow for a direct comparison to the performance of [Yao and Sun, 2016], in particular because their system has different goals.

Nevertheless, since GREN does use very related techniques, we believe it is useful as a reference baseline.

To ensure a fair comparison, and since we are focusing on the product linking task (as opposed to the mention recognition task), for the methods Wu and GREN only their solutions for the linking task were tested, and not their solutions for mention recognition. To this effect, when comparisons were performed, Wu and GREN were provided the exact same set of product mentions to be linked that was provided to ProdLink.

5.2.2 Dataset Analysis

Before presenting the results of our experiments, we first present an analysis of important characteristics of the ProdLink and CPROD1 datasets regarding the problem we address, such as mention and link statistics, and the kind mentions users typically make, including a measure of mention ambiguity.

Table 5.2 shows statistics on the mentions found in the datasets. For the ProdLink dataset, we show the statistics for each product category separately. In all labeled posts, there is a large amount of product mentions, although the CPROD1 dataset exhibits less mentions when compared to the ProdLink datasets. Also, the ProdLink dataset exhibits higher averages of different surface forms per product. On average, the datasets have 175 posts with mentions. Note that we do not consider cases where product mentions are made using pronouns, since the systems tested do not handle anaphora resolution.

As noted before, each product can have none or several different product mentions. In the BDP category of the ProdLink dataset, each product has, on average, 4.3 distinct surface forms. The average for CAM is 4.7, and 3.3 for SMRT. This demonstrates the diversity of the ways users refer to products.

Dataset/ Cat.	Product Mentions	Mentions w/ Links	Posts w/ Mentions	Avg. Mentions per Post	Avg. SF per Prod.
BDP	446	316(70.9%)	163(62.2%)	2.8	4.3
CAM	865	574(66.4%)	204(81.6%)	4.2	4.7
SMRT	860	602(70.0%)	225(90.0%)	3.8	2.9
CPROD1	463	186(40.2%)	109(5.2%)	4.2	1.1

Table 5.2: Mentions Statistics for the Datasets.

Dataset/ Cat.	Brand and Model	Brand	Model	Partial Model	Acron.
BDP	65(14.6%)	0(0.0%)	207(46.4%)	174(39.0%)	0(0.0%)
CAM	129(14.9%)	21(2.4%)	564(65.2%)	151(17.5%)	0(0.0%)
SMRT	38(16.0%)	0(0.0%)	453(52.7%)	221(25.7%)	48(5.6%)
CPROD1	259(55.9%)	4(0.9%)	176(38.0%)	20(4.3%)	4(0.9%)

Table 5.3: Mention characterization for the datasets.

In the CPROD1 dataset, each product has an average of 1.04 surface forms.

Table 5.2 shows in column "Mentions w/ Links" that not all mentions in the posts refer to a product in the catalog. Any suitable linking method should take this into account.

Further mention characterization is provided in Table 5.3. For each dataset we manually inspected each mention and grouped it into one of the following categories: brand and model, brand only, model only, partial model, and acronym. Partial model represents mentions where users employ part of the product model. One such example is "4" as referring to the *LG Nexus 4*. From the table we observe that the distribution of mention categories is different for each dataset, but is consistent in the different product categories in the ProdLink dataset. Model and partial model are prominent mention categories in the ProdLink dataset, but much less in the CPROD1 dataset. Acronyms are mostly used in the SMRT category, with just 4 uses in CPROD1.

Product mentions can also be ambiguous, i.e., a given surface form can potentially be linked to many products. To quantify this ambiguity, we use

Dataset/	Ambiguity	
Cat.	1	≥ 2
BDP	206 (65.2%)	110 (34.8%)
CAM	511 (89.0%)	63 (11.0%)
SMRT	486 (80.7%)	116 (19.3%)
CPROD1	58 (31.2%)	128 (68.8%)

Table 5.4: Mention ambiguity levels in the datasets.

the following definition: *given a single mention m in a post, count in the golden set the number of different products p to which m potentially refers to, that is, the number of products linked to mentions equal to m in all posts.* In Table 5.4, we present the distribution of the number of mentions per level of ambiguity. For example, in SMRT there are 116 mentions that can refer to at least 2 different products. Mentions with ambiguity greater than one represent on average approximately 33.5%. In the BDP category of the ProdLink dataset, in particular, these account for 34.8% of mentions. The CPROD1 dataset has a higher level of ambiguity because many products have duplicate entries in the catalog.

5.2.3 Evaluation

We start our evaluation by comparing ProdLink to both baseline systems. The ProdLink linker module was implemented through a Random Forest classifier using all features described in Section 5.1.4. The results shown are the cross-validation averages.

Since our focus is on the product linking task, and not on the product recognition task, in the following, we show the results for all systems when using the full set of manually labeled product mentions in the testing set. We note that this represents an upper bound on the performance of each product linking solution, since it assumes the mention recognizer module is perfect.

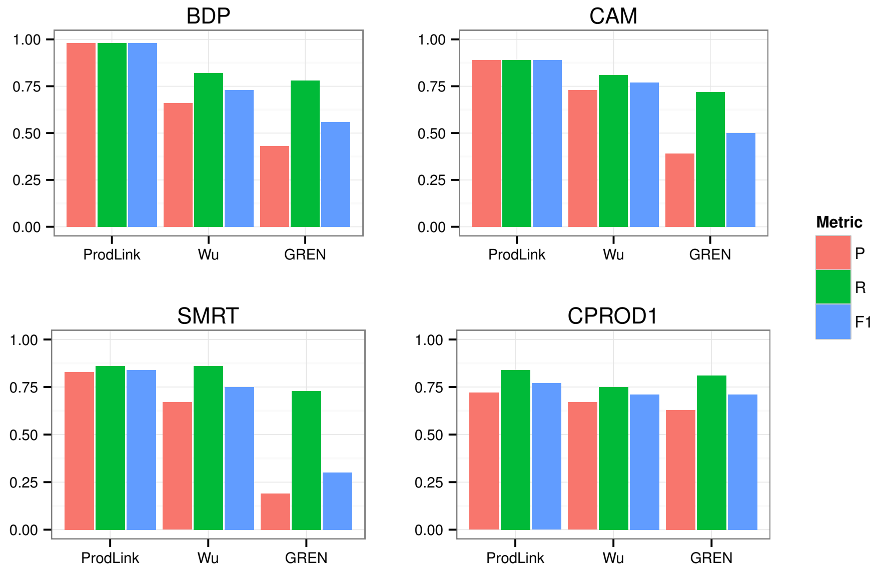


Figure 5.2: Precision, recall and F_1 results comparing ProdLink to the baseline systems.

Results achieved by the ProdLink system as an end-to-end solution are shown later, in Section 5.2.6.

Comparison to the Baselines

Figure 5.2 shows the results obtained by each product linking approach. We can see that ProdLink achieved higher values for precision, recall and F_1 in all datasets, when compared to Wu and GREN. On average, our method achieved gains over Wu of about 0.17 in precision, 0.08 in recall, and 0.13 in F_1. Our gains over GREN in precision, recall, and F_1 were about 0.45, 0.13, and 0.35, respectively. As the CPROD1 dataset contains less products that share the same surface form, ProdLink results are closer to the baselines.

It is also important to note that the gains in CPROD1 were achieved in spite of some labeling inconsistencies in the dataset that we opted not to correct. For example, many products that should be labeled as linked are not,

Method	BDP			CAM			SMRT			CPROD1		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
ProdLink	**0.94**	0.94	**0.94**	**0.77**	0.80	**0.78**	**0.74**	0.94	**0.83**	**0.52**	0.87	**0.65**
Wu	0.39	**1.00**	0.57	0.38	0.85	0.52	0.48	**0.96**	0.64	0.47	0.69	0.61
GREN	0.20	**1.00**	0.33	0.31	**0.88**	0.46	0.16	**0.96**	0.28	0.45	**0.93**	0.60

Table 5.5: Precision, recall and F_1 for ProdLink and the baselines when testing only on ambiguity ≥ 2 mentions.

which has a clear impact on precision levels. Another important aspect of the CPROD1 dataset is that it is actually a diverse set of catalogs from different sources. This scenario, along with the lack of contextual information, hinders classifiers from capturing strong signals that would otherwise enable better classification results.

Performance Analysis

It is interesting to take a deeper look at the reasons behind the results achieved. In first place, given the string matching strategy adopted by both Wu and GREN, we would expect recall levels closer to 100%. However, their greedy matching procedure penalizes both baselines for mentions that have no correspondence in the product catalog. Also, string matching decreases recall levels for mentions that specify only a brand and a partial model name. For example, using “Apple 5s” to reference the “Apple iPhone 5s” smartphone.

Another issue with partial product mentions, such as “4”, “5” and “105”, is that of ambiguity. Only by considering the context in which such mentions occur it is possible to correctly link these to their respective products. To test this hypothesis, we conducted an experiment where we only considered the mentions with ambiguity ≥ 2 (see Table 5.4). The results are shown in Table 5.5. It is clear that ProdLink achieves precision values much higher than the baselines, with a comparatively small loss in recall. Our average precision is 0.74, whereas the average precision of the best baseline is 0.43. Based on the context, ProdLink is able to correctly link ambiguous mentions to their proper catalog entry instead of assigning them to all offers that contain the corresponding tokens.

Finally, it is also worth mentioning the cases where none of the methods was successful at linking product mentions. One such case is related to different product generations that share a common string prefix. Take, for example, the product mention “Samsung Galaxy”. This mention corresponds to the original smartphone model, which is not present in the catalog. However, all solutions linked the mention to all the “Samsung Galaxy” products listed. In this case, not even context can be used as a clue, since it is similar to all mentions of all “Samsung Galaxy” versions. A similar error occurs when different products by the same manufacturer share a common prefix, distinguishing only at the end of the mention. For example, the mention “S4”, which is frequently linked to “Samsung Galaxy S4” and “Samsung Galaxy S4 Mini”, but should only be linked to the former.

Another case occurs because, in user-generated opinionated text, it is common for users to compare products. This leads to sentences such as “Well, I love my nexus 4, and would have sprung for the 5, but my wife got me an iphone 6”. In this example, “5” should be liked to “LG Nexus 5”, but all the tested approaches typically link the mention to “Apple iPhone 5”.

Ability to Generalize

An interesting aspect to observe in our results is how well ProdLink is able to link unknown surface forms for a given product, starting from known surface forms provided in the training set. To verify this, Table 5.6 shows the results when ProdLink was tested with mentions present only in the test set, i.e., new surface forms not encountered during training.

Dataset/Cat.	P	R	F_1
BDP	0.95	0.95	0.95
CAM	0.89	0.90	0.89
SMRT	0.84	0.84	0.84
CPROD1	0.72	0.85	0.78

Table 5.6: Results when testing only on unseen surface forms.

We can see that ProdLink achieves results similar to those achieved when all mentions are considered during testing. This is particularly evident in CPROD1, where most of the mentions in the test set were not present in the training set. This demonstrates that our method is able to generalize well for new data.

5.2.4 Selecting the Set of Features

Although using the full set of features described in Section 5.1.4 already provided strong results, it is important to verify if this is the case for all datasets. We performed a feature selection study. Specifically, we first group our proposed features according to the type of information they convey: Description Similarity features form group g_1, Positional features form group g_2, Acronym Similarity features form group g_3, and Context Similarity features form group g_4. We then run experiments considering each group in isolation and combinations of these groups, in a process of forward selection [Chandrashekar and Sahin, 2014], using F_1 as the criteria to select the best group. The selection process starts with a set containing only one group of features and adds other groups of feature to the best group combination. In Table 5.7 we show the results. Values in bold indicate the highest value achieved for each product category/dataset.

Interestingly, Table 5.7 shows that the combination of all feature groups indeed achieves the best results in all product categories. Nevertheless, we

Feature	BDP			CAM			SMRT			CPROD1		
Set	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$
g_1	0.81	**0.98**	0.89	0.86	0.86	0.86	0.78	0.86	0.82	0.59	0.77	0.67
g_2	0.27	0.27	0.27	0.79	0.79	0.79	0.81	0.88	0.84	0.41	0.50	0.45
g_3	0.29	0.29	0.29	0.42	0.42	0.42	0.42	0.46	0.44	0.41	0.41	0.41
g_4	0.32	0.35	0.33	0.29	0.30	0.29	0.21	0.21	0.21	0.39	0.43	0.40
g_1,g_2	0.81	**0.98**	0.89	0.86	0.86	0.86	0.79	0.85	0.81	0.59	0.77	0.67
g_1,g_3	0.81	**0.98**	0.89	0.88	0.88	0.88	0.81	0.84	0.82	0.59	0.77	0.67
g_1,g_4	0.97	0.97	0.97	**0.89**	**0.89**	**0.89**	0.76	0.80	0.78	**0.72**	**0.84**	**0.77**
g_1,g_2,g_3	0.81	**0.98**	0.89	0.88	0.88	0.88	0.79	0.85	0.81	0.59	0.77	0.67
g_1,g_2,g_3,g_4	**0.98**	**0.98**	**0.98**	**0.89**	**0.89**	**0.89**	**0.83**	**0.86**	**0.86**	**0.72**	**0.84**	**0.77**

Table 5.7: Feature selection results for ProdLink.

can also draw interesting conclusions from the remaining group combinations. We can see, for example, that group g_1 alone (description similarity) explains most of the success of ProdLink, since precision and recall values are much higher when compared to the other groups, and remain high when g_1 is in a combination.

The second most useful group is g_4 (contextual features). Although, in most cases, it yields lower values than g_2 or g_3 when in isolation, it achieves the highest gains in all measures when combined with g_1. The exception is the smartphone category, where group g_3 (acronym similarity) is also of importance. This confirms our hypothesis that contextual features are fundamental to achieve higher precision values.

In general, group g_2 (positional features) seems to be somewhat redundant when the remaining groups are present, while g_3 seems to be important only for smartphone mentions. This latter conclusion is confirmed by Table 5.3, where we can see that the usage of acronyms is prevalent mostly for the smartphone category.

5.2.5 Learning Algorithms for Product Linking

For completeness, we now present a study on the impact of using different learning algorithms for the ProdLink linker module. More specifically, we

Learning	BDP			CAM			SMRT			CPROD1		
Alg.	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
NB	0.65	0.86	0.74	0.63	0.80	0.70	0.47	0.84	0.60	0.64	0.82	0.72
DT	0.96	0.97	0.96	0.88	0.89	0.88	0.79	0.82	0.80	0.72	0.82	0.76
RF	**0.98**	**0.98**	**0.98**	**0.89**	**0.89**	**0.89**	**0.83**	0.86	0.84	**0.72**	**0.84**	**0.77**
SVM	0.93	0.93	0.93	0.87	0.87	0.87	**0.83**	**0.89**	**0.86**	0.71	0.81	0.76

Table 5.8: Results for different learning algorithms in the ProdLink Linker module.

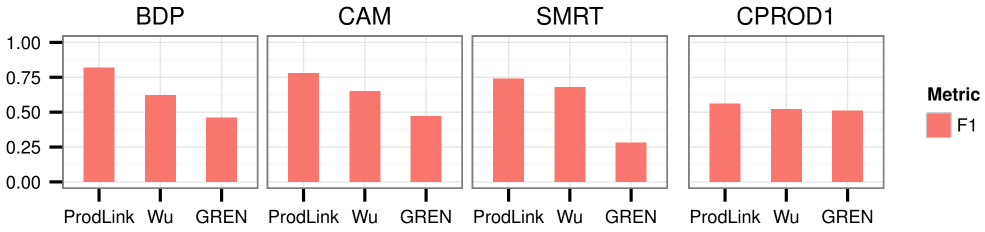


Figure 5.3: F_1 results using the Recognizer module.

tested Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), and SVM classifiers. Results are shown in Table 5.8. The values in bold indicate the highest value achieved for each algorithm.

We can see that Naive Bayes achieved the lowest results, while the remaining classifiers all had similar results. Since the Random Forest classifier presented the best overall behavior, we opted to use it in all the experiments.

5.2.6 Mention Recognition and Linking

Although in this work we are mostly interested on the performance of the Linker module, it is also important to evaluate the effect of the Recognizer module on the overall system performance.

Figure 5.3 shows the F_1 results obtained by each product linking approach when tested using mentions detected by the ProdLink Recognizer module. As expected, results are inferior to those shown in Figure 5.2, with a loss in F_1 of 0.16, 0.11, 0.10, and 0.21 in the BDP, CAM, and SMRT categories, and the CPROD1 dataset, respectively. We attribute the higher difference in

CPROD1 to labeling problems in the dataset, i.e., mentions detected by the Recognizer module without a corresponding product link.

Interestingly, if we compare the results of ProdLink in Figure 5.3, with the results achieved by the baselines in Figure 5.2, we notice that ProdLink results are still higher than those achieved by the baselines when tested using the set of manually labeled product mentions. Specifically, ProdLink shows an average F_1 value above those of Wu and GREN in 0.26 and 0.44 points, respectively, thus confirming its superior linking approach.

For reference, the Recognizer module of ProdLink achieved the following values of precision, recall and F_1 in each category/dataset — BDP: 0.78, 0.78, 0.78; CAM: 0.76, 0.74, 0.75; SMRT: 0.81, 0.79, 0.80; and CPROD1: 0.70, 0.71, 0.70.

5.3 Remarks

In this chapter, we presented a novel method to link product mentions, occurring in specialized discussion forums, to their respective real-world products, listed in a product catalog. This method was applied in a product linking system, called ProdLink, which is capable of both recognizing product mentions in natural language text and of linking those mentions the entries in a catalog. Our method makes use of machine learning techniques to combine a wide set of statistical features, in order to determine if a given mention should or not be linked to a given catalog entry.

Experiments were executed, not only to analyze the impact of such features on the performance of product linking, but also to compare our approach to two state-of-the-art solutions. Using ProdLink, we were indeed able to achieve values for precision, recall, and F_1 higher than both baselines. When compared to the best performing baseline, the gains obtained in precision, recall, and F_1 values were approximately 0.17, 0.08, and 0.13, respectively. We were also able to conclude that the set of features proposed for linking product mentions was quite adequate, with a consistent performance between the different datasets. In particular, we show that features that exploit contextual information are fundamental to achieve high precision results. All experiments were performed using two datasets. In sum, the contributions of this work are: a novel and effective method for the problem of product linking, a set of features geared toward product mention disambiguation, a characterization of how users mention products, and a new dataset for product linking research.

Chapter 6

Conclusions and Future Work

In this thesis, we presented contributions to address the problem of using the unstructured textual content generated by social media users for extracting and categorizing opinion target entities. In our work, we focused on target entities of a specific, and relevant, type: consumer electronic products, such as smartphones, digital cameras and Blu-ray players. The task we addressed here is how to recognize and link mentions in user generated textual content to the product, from a catalog, they refer to. Ultimately this allows the application of opinion mining techniques, and continuously enriching a knowledge about products represented in product catalogs.

We formalized the *product recognizing and linking* task as the process of automatically associating a mention to a product in a text document or fragment to an entry representing that product in a catalog. We approached the task with two basic sub-tasks: (a) automatically identifying a mention m to a product in a text document or fragment; and (b) automatically associating product mention m to an entry representing product p in a catalog C.

As our first contribution to the sub-task of recognizing product mentions from unstructured textual content was ModSpot (Product **Mod**el Number

Spotter), a method for learning a CRF to undertake the task of identifying model numbers of products of a given category. The method is based on a self-training process that requires only a set of initial seed model numbers from consumer products, which means it does not require annotated training sentences to be provided. Experiments in four settings demonstrated that our method achieved similar or better results when compared to a supervised CRF with the same feature set. All the experimented settings exhibited higher F-measures when our process finished, and the seed set is about 40% larger. In particular, the expansion in seeds performed by the method helped to achieve higher recall levels. In addition, our method converged at around 9-14 iterations, when ModSpot could not identify new seeds.

Although unsupervised and requiring only a set of seed examples, our first contribution to the sub-task of recognizing product mentions from unstructured textual content was limited to product model numbers. To overcome this limitation, we proposed a new method, called ProdSpot. In a nutshell, ProdSpot goes through following steps. Initially, typical surface forms used as mentions to products are extracted from a set of product descriptions. Then, given a collection of user posts, the method identifies sentences that contain the extracted surface forms. To improve the quality of the extracted mentions, a cluster-based filtering strategy is applied, to detect and filter out possible false examples, which could compromise the precision of the generated model. Finally, to avoid overfitting, our method uses the initial set of sentences to produce more general and diverse set of synthetic sentences. It is this final set of synthetic sentences that will constitute the training set for learning a product mention recognition model. ProdSpot achieved competitive values in all categories when compared to a supervised CRF model, despite being distantly supervised and not requiring any user input.

For the second sub-task, we presented a novel method to link product mentions, occurring in specialized discussion forums, to their respective real-world products, listed in a product catalog. This method was applied in a product linking system, called ProdLink, which is capable of both recognizing product mentions in natural language text and of linking those mentions the entries in a catalog. Our method makes use of machine learning techniques to combine a wide set of statistical features, in order to determine if a given mention should or not be linked to a given catalog entry.

Experiments were executed, not only to analyze the impact of such features on the performance of product linking, but also to compare our approach to two state-of-the-art solutions. Using ProdLink, we were indeed able to achieve values for precision, recall, and F_1 higher than both baselines. When compared to the best performing baseline, the gains obtained in precision, recall, and F_1 values were approximately 0.17, 0.08, and 0.13, respectively. We were also able to conclude that the set of features proposed for linking product mentions was quite adequate, with a consistent performance between the different datasets. In particular, we show that features that exploit contextual information are fundamental to achieve high precision results. All experiments were performed using two datasets, one of which was created by the authors and made publicly available to the community. In sum, the contributions of this work are: a novel and effective method for the problem of product linking, a set of features geared toward product mention disambiguation, a characterization of how users mention products, and a new dataset for product linking research.

6.1 Future Work

The results we have achieved with the work presented here opens a number of possible ideas for future development.

To further enhance our methods and its initial bootstrapping seed set, it would be interesting to investigate techniques such as set expansion or one-class classifier as means to have more seeds during the automatically annotation of the classifier training set. We expect that such expanded seeds would allow better classification models, with a more diverse training set.

Automatic identification of brand names mentions that are non-product mention examples should further enhance the training set produced by ProdSpot. This future work should aim at reducing the mislabeling errors attributed to tokens which are brand names, and help classifier generalization. Experimental results have demonstrated that the classification models tended to mislabel brand name tokens when used to mention the actual brand.

As our contributions to the sub-task of recognizing product mentions from unstructured textual content are distantly supervised, the CRF classifier used in the methods might be replaced by other classifiers. One interesting work could investigate the use of deep learning architectures [Lample et al., 2016] as a product mention classifier.

Although effective, our contribution to mention linking was based on a supervised training set. An issue we did not address in our work is automatically generating training data to a linking classification model in a distant supervised approach.

An very interesting future work is jointly performing product mention recognition and linking in a single unified framework. Such framework should take as input a product catalog and target social media content to enable product mention recognition and ultimately link mentions to products.

In our work, we focused on target entities of a specific type: consumer electronic products. One interesting direction is the application of the methods presented here to other types of target entities, such as books, movies, hotels and restaurants. We expect that other entity types require changes to the Seeding and Training Sentence Synthesis steps to account for how these new entities are typically listed in a catalog, and how users mention these entities in text.

Finally, once mentions from social media are recognized and linked to their respective products in a catalog, a framework could automatically extract and map user opinions from social media contents to the product attributes from the catalog.

6.2 Publications

Following we list all publications produced during this work.

Vieira, Henry S., Altigran S. da Silva, Marco Cristo, and Edleno S. de Moura. A self-training CRF method for recognizing product model mentions in web forums. *In European Conference on Information Retrieval*, pp. 257-264. Springer, 2015.

Vieira, Henry S., Altigran S. da Silva, Pável Calado, Marco Cristo, and Edleno S. de Moura. Towards the effective linking of social media contents to products in e-commerce catalogs. *In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1049-1058. ACM, 2016.

Vieira, Henry S., Altigran S. da Silva, Pável Calado, and Edleno S. de Moura. A distantly supervised approach for recognizing product mentions in user-generated content. *Submitted to IEEE Transactions on Knowledge and Data Engineering.*

Bibliography

[Baeza-Yates and Ribeiro-Neto, 2011] Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (2011). *Modern Information Retrieval - the concepts and technology behind search.* Pearson Education, 2nd edition.

[Breck and Cardie, 2017] Breck, E. and Cardie, C. (2017). Opinion mining and sentiment analysis. In *The Oxford Handbook of Computational Linguistics.* Oxford University Press, 2nd edition.

[Brown et al., 1992] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

[Castellanos et al., 2011] Castellanos, M., Dayal, U., Hsu, M., Ghosh, R., Dekhil, M., Lu, Y., Zhang, L., and Schreiman, M. (2011). Lci: a social channel analysis platform for live customer intelligence. In *Proceedings of the 2011 ACM SIGMOD*, pages 1049–1058.

[Ceccarelli et al., 2013] Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., and Trani, S. (2013). Learning relatedness measures for entity linking. In *Proceedings of the 2013 ACM international conference on Information and Knowledge Management*, pages 139–148.

[Chandrashekar and Sahin, 2014] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40:16–28.

[Chawla et al., 2017] Chawla, S., Dubey, G., and Rana, A. (2017). Product opinion mining using sentiment analysis on smartphone reviews. In *Proceedings of the 2017 International Conference on Reliability, Infocom Technologies and Optimization*, pages 377–383.

[Chen et al., 2008] Chen, M., Chen, Y., and Brent, M. R. (2008). Crf-opt: An efficient high-quality conditional random field solver. In *Proceedings of the 2008 AAAI*, pages 1018–1023.

[Choi and Lee, 2017] Choi, B. and Lee, I. (2017). Trust in open versus closed social media: The relative influence of user-and marketer-generated content in social network services on customer trust. *Telematics and Informatics*, 34:550–559.

[Cucerzan, 2007] Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.

[Culotta and McCallum, 2004] Culotta, A. and McCallum, A. (2004). Confidence estimation for information extraction. In *Proceedings of the 2004 HLT-NAACL*, pages 109–112.

[Dalvi et al., 2009a] Dalvi, N., Kumar, R., Pang, B., and Tomkins, A. (2009a). Matching reviews to objects using a language model. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 609–618.

[Dalvi et al., 2009b] Dalvi, N., Kumar, R., Pang, B., and Tomkins, A. (2009b). A translation model for matching reviews to objects. In *Proceedings of the 2009 ACM International Conference on Information and Knowledge Management*, pages 167–176.

[Derczynski et al., 2017] Derczynski, L., Nichols, E., van Erp, M., and Limsopatham, N. (2017). Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 2017 Workshop on Noisy User-generated Text*, pages 140–147.

[Dredze et al., 2010] Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 2010 international conference on Computational Linguistics*, pages 277–285.

[Eisenstein, 2013] Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.

[Elmagarmid et al., 2007] Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate record detection: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 19:1–16.

[eMarketer, 2014] eMarketer (2014). Global b2c ecommerce sales to hit $1.5 trillion this year driven by growth in emerging markets. `http://goo.gl/8VMRgz`. Accessed May/2015.

[Feldman, 2013] Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

[Florian et al., 2003] Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the 2003 Conference on Natural Language Learning at HLT-NAACL*, pages 168–171.

[Frénay and Verleysen, 2014] Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

[Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 1996 Conference on Computational Linguistics*, pages 466–471.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, 2 edition.

[He and Garcia, 2008] He, H. and Garcia, E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, pages 1263–1284.

[Hobbs and Riloff, 2010] Hobbs, J. R. and Riloff, E. (2010). Information extraction. *Handbook of natural language processing.*

[Jakob and Gurevych, 2010] Jakob, N. and Gurevych, I. (2010). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Empirical Methods in Natural Language Processing*, pages 1035–1045.

[Ji et al., 2017] Ji, H., Pan, X., Zhang, B., Nothman, J., Mayfield, J., McNamee, P., Costello, C., and Hub, S. I. (2017). Overview of tac-kbp2017

13 languages entity discovery and linking. In *Proceedings of the 2017 Text Analysis Conference*.

[Kaplan and Haenlein, 2010] Kaplan, A. M. and Haenlein, M. (2010). Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59–68.

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 2001 International Conference on Machine Learning*, pages 282–289.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

[Lavergne et al., 2010] Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale CRFs. In *Proceedings the 2010 Association for Computational Linguistics*, pages 504–513.

[Li et al., 2013] Li, Y., Wang, C., Han, F., Han, J., Roth, D., and Yan, X. (2013). Mining evidences for named entity disambiguation. In *Proceedings of the 2013 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1070–1078.

[Liao and Veeramachaneni, 2009] Liao, W. and Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition. In *Proceedings the 2009 NAACL HLT Workshop SSLNLP*, pages 58–65.

[Liu, 2007] Liu, B. (2007). *Web Data Mining*. Springer.

[Liu, 2012] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, pages 1–167.

[Melli, 2014] Melli, G. (2014). Shallow semantic parsing of product offering titles (for better automatic hyperlink insertion). In *Proceedings of the 2014 ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 1670–1678.

[Melli and Romming, 2012] Melli, G. and Romming, C. (2012). An overview of the CPROD1 contest on consumer product recognition within user generated postings and normalization against a large product catalog. In *Proceedings of the 2012 IEEE ICDM Workshops*, pages 861–864.

[Moghaddam and Ester, 2013] Moghaddam, S. and Ester, M. (2013). Opinion mining in online reviews: Recent trends. Tutorial at WWW2013.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

[Navigli, 2009] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10:1–10:69.

[Niyogi et al., 1998] Niyogi, P., Girosi, F., and Poggio, T. (1998). Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 86(11):2196–2209.

[Pang and Lee, 2008] Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, pages 1–135.

[Peng and Dredze, 2015] Peng, N. and Dredze, M. (2015). Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

[Penn and Zalesne, 2013] Penn, M. and Zalesne, E. K. (2013). New info shoppers. the wall street journal. Web page retrieved in June 27th 2013 and available at http://online.wsj.com/article/SB123144483005365353.htm.

[Poria et al., 2016] Poria, S., Cambria, E., and Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

[Presi et al., 2014] Presi, C., Saridakis, C., and Hartmans, S. (2014). User-generated content behaviour of the dissatisfied service customer. *European Journal of Marketing*, 48:1600–1625.

[Putthividhya and Hu, 2011] Putthividhya, D. P. and Hu, J. (2011). Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Empirical Methods in Natural Language Processing*, pages 1557–1567.

[Rao et al., 2013] Rao, D., McNamee, P., and Dredze, M. (2013). Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 93–115. Springer.

[Ratinov and Roth, 2009] Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the 2009 Computational Natural Language Learning*, pages 147–155.

[Rau, 1991] Rau, L. F. (1991). Extracting company names from text. In *Proceedings of the 1991 IEEE Conference on Artificial Intelligence Applications*, pages 29–32.

[Santosh et al., 2016] Santosh, D. T., Babu, K. S., Prasad, S., and Vivekananda, A. (2016). Opinion mining of online product reviews from traditional lda topic clusters using feature ontology tree and sentiwordnet. *International Journal of Education and Management Engineering*, 6:34–44.

[Sarawagi, 2008] Sarawagi, S. (2008). Information extraction. *Foundations and Trends in Databases*, 1(3):261–377.

[Sekine and Nobata, 2004] Sekine, S. and Nobata, C. (2004). Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the 2004 Language Resources and Evaluation Conference*, pages 1977–1980.

[Sethna et al., 2017] Sethna, B. N., Hazari, S., and Bergiel, B. (2017). Influence of user generated content in online shopping: impact of gender on purchase behaviour, trust, and intention to purchase. *International Journal of Electronic Marketing and Retailing*, 8:344–371.

[Shen et al., 2015] Shen, W., Wang, J., and Han, J. (2015). Entity linking with a knowledge base: issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27:443–460.

[Shen et al., 2012] Shen, W., Wang, J., Luo, P., and Wang, M. (2012). Linden: linking named entities with knowledge base via semantic knowledge. In

Proceedings of the 2012 international conference on World Wide Web, pages 449–458.

[Song et al., 2004] Song, Y., Kim, E., Lee, G. G., and Yi, B.-k. (2004). Posbiotm-ner in the shared task of bionlp/nlpba 2004. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 100–103.

[Sutton and McCallum, 2012] Sutton, C. A. and McCallum, A. (2012). An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

[Teixeira et al., 2011] Teixeira, J., Sarmento, L., and Oliveira, E. (2011). A bootstrapping approach for training a ner with conditional random fields. In *Proceedings of the 2011 EPIA*, pages 664–678.

[Tjong Kim Sang and De Meulder, 2003] Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 2003 conference on Natural Language Learning at HLT-NAACL*, pages 142–147.

[Vlachos and Gasperin, 2006] Vlachos, A. and Gasperin, C. (2006). Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 138–145.

[Wu et al., 2012] Wu, S., Fang, Z., and Tang, J. (2012). Accurate product name recognition from user generated content. In *Proceedings of the 2012 IEEE ICDM Workshops*, pages 874–877. IEEE.

[Yao and Sun, 2016] Yao, Y. and Sun, A. (2016). Mobile phone name extraction from internet forums: a semi-supervised approach. *World Wide Web*, 19:783–805.

[Yarowsky, 1995] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 1995 annual meeting on Association for Computational Linguistics*, pages 189–196.

[Zhang and Liu, 2011] Zhang, L. and Liu, B. (2011). Entity set expansion in opinion documents. In *Proceedings of the 2011 ACM HT*, pages 281–290.

[Zhang et al., 2011] Zhang, W., Sim, Y. C., Su, J., and Tan, C. L. (2011). Entity linking with effective acronym expansion, instance selection, and topic modeling. In *IJCAI*, volume 2011, pages 1909–1914.

[Zhu and Wu, 2004] Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210.