Methoden voor efficiënte supervisie van automatische taalverwerking

Methods for Efficient Supervision in Natural Language Processing

Lucas Sterckx

Promotoren: prof. dr. ir. C. Develder, dr. ir. T. Demeester
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

UNIVERSITEIT
GENT

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. B. Dhoedt
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2017 - 2018

ISBN 978-94-6355-130-4
NUR 984
Wettelijk depot: D/2018/10.500/48

Ghent University
Faculty of Engineering and Architecture
Department of Information Technology

imec
Internet Technology and Data Science Lab

Methods for Efficient Supervision in Natural Language Processing

Examination Board:

prof. C. Develder (advisor)
dr. ir. T. Demeester (advisor)
em. prof. D. De Zutter (chair)
prof. B. Dhoedt (secretary)
prof. I. Augenstein
prof. A. Bronselaer
prof. V. Hoste

Dissertation for acquiring the degree of
Doctor of Computer Science Engineering

Dankwoord

Tijdens mijn doctoraat werd ik onvoorwaardelijk, door dik en dun, gesteund door vele mensen. Ik dank iedereen die me alle kansen gaf tot persoonlijke en professionele groei. Een dankjewel,

aan prof. Chris Develder, die me vijf jaar lang het vertrouwen gaf en alle kansen tot wetenschappelijk onderzoek, samen met de vrijheid tot het bewandelen van mijn eigen pad;

aan Thomas Demeester, die, ondanks zijn eigen drukke agenda, altijd tijd maakte en klaar stond met advies en een luisterend oor, steeds met hetzelfde enthousiasme en dezelfde vriendelijkheid;

aan mijn ouders, Godelieve en Paul;

aan mijn broer en schoonzus, Thomas en Karolien;

aan mijn meter, Maria;

aan Johannes, die me met al zijn geduld en kennis bijstond vanaf dag één en me leerde kritisch zijn over eigen en bestaand werk;

to Nasrin, who, despite living abroad, was always cheerful and kind to everyone, all while producing an incredible amount of high-quality research;

to Giannis, or Lucas *V*2.0 as I like to call him, who picked up the torch and ran so fast with it, for making the future of research in NLP at IDLab look bright;

to all my research collaborators over the years, Cedric, Fréderic, Baptist, Klim, Thong, Steven, Matthias, Jason, Bill, Cornelia and Laurent, from whom I learned so much;

to the members of the examination board, em. prof. Daniël De Zutter, prof. Bart Dhoedt, prof. Isabelle Augenstein, prof. Antoon Bronselaer and prof. Veronique Hoste, for making time and providing me with nothing but constructive feedback on this thesis and my research in general;

aan prof. Piet Demeester, de drijvende kracht achter de onderzoeksgroep;

to the founders and attendees of the machine learning reading group, who gracefully shared their knowledge every week;

to the IDLab admins and support staff for always standing by with technical support and providing me with the infrastructure to do research;

to all my friends and colleagues at IDLab for providing me with such a pleasant working environment;

to all friends I made abroad, who made me feel at home when I was far from it;

aan alle collega's en jobstudenten met wie ik samen aan projecten werkte, om me de middelen te geven om aan onderzoek te doen;

aan het Fonds voor Wetenschappelijk Onderzoek - Vlaanderen (FWO), welke via een reisbeurs mijn onderzoek ondersteunde;

aan al mijn collega-muzikanten en de leden van Concertband Theobaldus Groot-Pepingen;

en aan al mijn vrienden uit het Pajottenland en het Gentse.

Gent, juni 2018
Lucas Sterckx

"The really important kind of freedom involves attention,
and awareness, and discipline, and effort,
and being able truly to care about other people and to sacrifice for them,
over and over, in myriad petty little unsexy ways, every day."

— David Foster Wallace, 2005

Table of Contents

List of Figures

List of Tables

List of Acronyms

ALA	Automated Lyric Annotation
AE	Autoencoder
AKE	Automated Keyphrase Extraction
CRF	Conditional Random Field
CNN	Convolutional Neural Network
DS	Distant Supervision
IE	Information Extraction
KB	Knowledge Base
KBP	Knowledge Base Population
LDA	Latent Dirichlet Allocation
LR	Logistic Regression
LSTM	Long Short-Term Memory network
ML	Machine Learning
NMT	Neural Machine Translation
NER	Named Enitity Recognition
NLP	Natural Language Processing
NN	Neural Network
POS	Part-of-Speech
RNN	Recurrent Neural Network
Seq2seq	Sequence-to-Sequence
SSL	Semi-Supervised Learning
SVM	Support Vector Machine

TAC	Text Analysis Conference
VAE	Variational Autoencoder
XGBoost	Extreme Gradient Boosting

Samenvatting
– Summary in Dutch –

Technieken voor artificiële intelligentie hebben een enorm potentieel om via innovatieve toepassingen ons leven comfortabeler te maken. Het onderzoeksdomein van natuurlijke taalverwerking of computerlinguïstiek stelt systemen voor AI ertoe instaat om taal te interpreteren and taken uit te voeren zoals entiteitsherkenning, extractie van relaties of sentiment detectie. Veel systemen voor taalverwerking vandaag zijn gebaseerd op machinaal lerende systemen die getraind worden met gelabelede, tekstuele data. Deze methodes leren autonoom van data en detecteren de onderliggende structuur en patronen. De hoeveelheid ongelabelde linguïstische data is echter veel groter dan de gelabelede en groeit veel sneller dan de hoeveelheid gelabelede tekst. De vraag naar methodes die efficiënt aan machinaal leren doen is daarom groot, i.e., methodes die met slechts kleine hoeveelheden gelabelede data complexe modellen kunnen trainen. Deze thesis focust op efficiënt gesuperviseerd machinaal leren voor een selectie van toepassingen binnen het domein van natuurlijke taalvewerking. We tonen hoe, via sterk gereduceerde manuele supervisie, we performante systemen kunnen bouwen voor geautomatiseerde extractie van informatie, evaluatie van topic modellen, extractie van keyphrases en sequentie-naar-sequentie modellen.

In een eerste applicatie bestuderen we automatische populatie van kennisdatabanken met feiten. Omdat feiten in kennisdatabanken vaak manueel moeten worden toegevoegd, zijn ze al snel onvolledig en niet up-to-date volgens de laatste informatie. Een systeem dat automatisch informatie extraheert uit allerlei soorten ongestructureerde data is daarom noodzakelijk. We kunnen beroep doen op technieken uit natuurlijke taalverwerking voor automatische aanvulling van deze kennisdatabanken uit tekst. Een belangrijke component van zulke systemen is de relatie extractor welke relaties detecteert tussen entiteiten in de tekst. Het trainen van zulke extractors vergt doorgaans echter grote hoeveelheden van voorbeelden van elke relatie in de kennisdatabase die automatisch dient te worden aangevuld (e.g., is_getrouwd_met, is_werknemer_van). We tonen dat de hoeveelheid benodigde gelabelede data sterk gereduceerd kan worden door gebruik te maken van zwakke supervisie, welke training data genereert door feiten

uit een bestaande kennisdatabase te aligneren met een grote collectie van tekst. Dit resulteert doorgaans in training data van lage kwaliteit, waar veel zinnen niet de relatie uitdrukken voor de welke ze gegenereerd werden. We introduceren een nieuwe techniek genaamd *semantische label propagatie* waarvoor we laag-dimensionale representaties gebruiken van het kortste-pad in de afhankelijkheid-graaf tussen de entiteiten, om zo data voor de relatie te bootstrappen. We tonen dat, met slechts enkele minuten van annotatie werk, we in staat zijn om de precisie van de relatie extractors sterk te vergroten. Door deze techniek toe te passen in een benchmark voor informatie extractie systemen van kennisdatabanken, zijn we in staat een hoge precisie score te behalen, hoger dan systemen die voor de benchmark gebruik maakten van complexere modellen en ensemble technieken.

Als tweede applicatie presenteren we een efficiënte methode om ongesuperviseerd, topicmodellen te evalueren. Topicmodellen werken op grote hoeveelheden tekst en geven ons een overzicht van de thematische inhoud aan de hand van topics of verzamelingen van woorden. Hoewel deze modellen statistisch onderbouwd zijn, bieden ze geen garantie om topics te genereren die interpreteerbaar zijn. Bestaande evaluatiemethoden om de kwaliteit van een topicmodel te meten zijn gebaseerd op statistische methodes of vragen gebruikers om alle topics handmatig te scoren. We stellen een nieuwe meettechniek voor die gebaseerd is op het gebruik van bestaande, kleinere collecties van gecategoriseerde tekst. Door de similariteit te meten tussen ongesuperviseerde en gesuperviseerde topics kunnen we scores aan de topics toekennen die veel sterker gecorreleerd zijn met manuele scores dan de bestaande methodes.

Automatische extractie van keyphrases is het automatisch samenvatten van lange documenten aan de hand van een selectie van korte zinnen die de inhoud van het volledige document kort samenvatten. Evaluatie van systemen voor extractie van keyphrases vereist grote collecties van documenten die voorzien zijn van keyphrases door meerdere annotatoren. Zulke collecties zijn lang niet beschikbaar geweest voor onderzoek. We organiseerden daarom de annotatie van collecties van nieuws, sportartikels en modeartikels door een divers panel van lezers en professionele schrijvers. Voordien was ook weinig consensus over de performantie van de verschillende methodes voor automatische extractie. We doen een systematische vergelijking tussen de bestaande ongesuperviseerde en gesuperviseerde technieken voor de test collecties. Nadien herformuleren we het probleem van keyphrase extractie als positief-ongelabelde classificatie, een vorm van semi-gesuperviseerde classificatie waarin we gelabelede keyphrases als positieve data zien en andere kandidaten zien als ongelabeled. Het gebruik van meerdere opinies voor keyphrase extractie lijdt tot betere modellen maar is duur. We stellen een procedure voor die, gebruik makende van slechts een enkele opinie, in staat is om gelijkaardige precisie te behalen dan de duurdere training data. Deze methode behaalt ook betere scores op andere, bestaande test collecties.

Tot slot stellen we twee applicaties voor van sequentie-naar-sequentie modelering via training op data van lage kwaliteit of inconsistente aligner-ing. Sequentie-naar-sequentie modellen zijn een van de belangrijkste toepassingen van deep learning modellen voor natuurlijke taalverwerking, met state-of-the-art performantie voor automatische vertaling. Deze modellen zijn vaak gebaseerd op recurrente neurale netwerken welke een groot aantal parameters bevatten die doorgaans dus grote hoeveelheden training data nodig hebben. We presenteren twee applicaties waarvoor grote collecties van training data voorheen niet beschikbaar waren. In een eerste setting stellen we een nieuwe taak, automatische lyric verklaring, voor met een bijhorende dataset die een groot aantal lyrics samen met bijhorende verklaringen bevat. Het doel van deze taak is om poëzie en straattaal te vertalen, een soort tekst waar bestaande systemen voor taalverwerking grote moeilijkheden ondervinden. De uitgebrachte dataset is de grootste van zijn soort en zal onderzoek in tekst normalisatie, verwerking van poëzie, en generatie van parafrasering stimuleren. In een tweede bijdrage, presenteren we een mechanisme om stijl-attributen van output-sequenties te controleren in sequentie-naar-sequentie modellen, via attentie-waardes en een verborgen latente vector, gegenereerd voor vertaling. We demonstreren dit principe voor automatische tekst simplificatie en tonen dat, door de latente vector waardes uit specifieke regionen te kiezen, we de lengte en alignering van output-sequenties kunnen controleren.

We geloven dat efficiënte supervisie in combinatie met deep learning modellen veel potentieel heeft en we verwachten dat in de toekomstige meer nieuwe applicaties gebruiken zullen maken van zwakke supervisie of gebruik zullen maken van efficiëntere, flexibelere supervisie. Dit zal uiteindelijk zorgen voor meer verspreide toepassing van natuurlijke taalverwerking en machinaal leren.

Summary

Artificial intelligence technologies offer great potential for creating new and innovative solutions to improve our daily lives. Natural language processing (NLP) enables artificial intelligence systems to interpret human language and perform tasks such as automatic summarization, translation, dialogue, named entity recognition, relationship extraction, and sentiment analysis. Many effective modern NLP systems are built using *supervised* machine learning methods. These methods learn from the data, by detecting and extracting underlying patterns and structure from examples, and rely on labeled training data. However, the amount of unlabeled linguistic data available to us is much larger and growing much faster than the amount of labeled data. Recent efforts in machine learning have addressed the increasing need for data-efficient machine learning: the ability to learn in complex domains without requiring large quantities of labeled data.

In this thesis we emphasize the use of efficient supervised learning for a selection of core tasks within NLP by making use of a variety of techniques which reduce the need for large quantities of supervision. We show how to build effective models from limited training data while still reaching state-of-the-art performance for information extraction, topic model evaluation, automatic keyphrase extraction and sequence-to-sequence modelling.

In a first application we present knowledge base population (KBP), the process of populating a knowledge base (KB), i.e., a relational database storing factual information, from unstructured and/or structured input, e.g., text, tables, or even maps and figures. Because KBs are often manually constructed, they are incomplete and not up-to-date with the latest information. KBP systems are crucial to keep KBs up-to-date by extracting information from unstructured data such as webtext. Relation extractors are important components of KBP systems. Training relation extractors for the purpose of automated knowledge base population requires the availability of sufficient labeled training data for each predicate or relation in the KB (e.g., *spouse_of*, *top_member_of*). We show that the amount of manual labeling can be significantly reduced by first applying distant supervision, which generates training data by aligning large text corpora with existing knowledge bases. However, this typically results in a highly noisy training set, where many training sentences do not express the intended relation. We introduce a technique called *semantic label propagation* in which

we use low dimensional representations of shortest dependency paths between entities of interest to bootstrap classifiers. We show that, with only minutes of labeling per relation we are able to match or improve on the accuracy obtained by fully supervised relation extractors. By including this technique in a KBP system, we achieved top-ranking submissions to a shared task for KBP systems. Using more but less noisy training data, our sparse-feature-based linear classifiers were able to obtain higher accuracies than systems using more sophisticated ensembles and deep learning architectures.

In a second application, we present an efficient method for the evaluation of topic models by making use of small labeled text collections. State-of-the-art unsupervised topic models lead to reasonable statistical models of documents but offer no guarantee of creating topics that are interpretable by humans. A careful evaluation requires manual supervision which can be costly for large topic models. Instead, we use existing, smaller labeled text collections to provide us with reference concepts and present a new measure for topic quality based on alignment between these supervised and unsupervised topics. Our proposed measure was shown to correlate better with human evaluation than existing unsupervised evaluation measures.

Automatic keyphrase extraction is the task of automatically extracting the most important and topical phrases of a document. Proper evaluation of keyphrase extraction requires large test collections with multiple opinions which were not available for research. We developed large corpora of news, sports and fashion articles annotated with keyphrases by a diverse crowd of laymen and professional writers. Prior, there was little consensus on the definition of the task of keyphrase extraction, few large benchmark collections of keyphrase-labeled data, and a lack of overview of the effectiveness of different techniques. We benchmark existing techniques for supervised and unsupervised keyphrase extraction on the newly introduced corpora. Next to benchmarking existing techniques, we study the influence of overlap in the annotations on the performance metrics. We rephrase the supervised keyphrase extraction problem as positive unlabeled learning in which a binary classifier is learned in a semi-supervised way from only positive keyphrases and unlabeled candidate phrases. The use of multiple annotations leads to more robust automatic keyphrase extractors, we propose reweighting of labels by a single annotator, based on probabilities by a first-stage classifier. This reweighting approach outperforms other state-of-the-art automatic keyphrase extractors using a single opinion on different test collections.

As a final contribution we present two applications of sequence-to-sequence models trained on noisy or poorly aligned training data. Sequence-to-sequence models are one of the most impactful applications of deep learning architectures to NLP, providing state-of-the-art results for machine translation. These architectures, mostly relying on recurrent neural

networks are heavily parameterized and require large amounts of high-quality training data. We present two applications of sequence-to-sequence learning using weak supervision. We study two applications where only noisy or low quality training data is available. In a first setting we present the novel task of automated lyric annotation and an accompanying dataset providing explanations to lyrics and poetic text. The goal of this task is generating explanations to poetic and slang text, a type text which existing NLP systems have great difficulty with. The presented dataset is one of the largest of its kind and will stimulate research in text normalization, metaphor processing and paraphrase generation. In a second contribution, we extend sequence-to-sequence models with the possibility to control the characteristics or style of the generated output, via attention that is generated a priori (before decoding) from a latent code vector space. After training an initial attention-based sequence-to-sequence model, we use a variational autoencoder conditioned on representations of input sequences and a latent code vector to generate attention matrices. By sampling the code vector from specific regions of a latent space during decoding and imposing prior attention in the seq2seq model, output can be steered towards having certain attributes. This was demonstrated for the task of sentence simplification, where the latent code vector allows control over output length and lexical simplification, and enables fine-tuning to optimize for different evaluation metrics.

We believe applications of reduced supervision in combination with deep learning models offer great potential and we expect weak supervision approaches will continue to be translated into more efficient, more flexible, and eventually more usable systems for NLP and machine learning.

1 Introduction

"Content without method leads to fantasy,
method without content to empty sophistry."

— Johann Wolfgang von Goethe [1]

Artificial Intelligence (AI) technologies offer great potential for creating new and innovative solutions to improve people's lives, address challenges in health and wellbeing, climate change, safety and security. In recent years AI has seen incredible progress and increasingly found its way into our daily lives. Rather than programming computers explicitly for a specific task, a shift towards data-driven approaches has taken place.

Natural Language Processing (NLP) is the field that focuses on the interactions between human language and computers. It is situated at the intersection of Artificial Intelligence, computer science, and computational linguistics. NLP is a way for computers to analyze, understand, and derive meaning from human language, and enables AI systems to perform tasks such as automatic text summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

Early attempts to solve NLP tasks relied on expensive hand-engineered manual rules. Such rule-based systems suffer from many drawbacks such as cost and difficulty to train experts in order to define and maintain the rules [2]. This is why today, many effective modern NLP systems are achieved using statistical machine learning methods.

Machine learning studies artificial systems that learn from data. Machine learning is critical for AI applications such as computer vision, data mining, clinical decision support systems and NLP. Because machine learning systems learn from data by detecting and extracting underlying patterns and structure from examples, they provide potential solutions for many of the problems faced in these domains.

Most of the successful applications of machine learning techniques used for NLP are currently supervised methods, which rely on labeled training data. However, the amount of unlabeled linguistic data available to us is much larger and growing much faster than the amount of labeled data which expensive and slow to generate. This is why recent efforts in machine learning have addressed the increasing need for supervision-efficient machine learning: the ability to learn in complex domains without requiring large quantities of labeled data. In the past, techniques such as semi-supervised learning, active learning, transfer learning, multitask learning addressed this need and searched for ways to build better models at lower cost. This thesis presents several applications of data-efficient supervised learning for a selection of core tasks within NLP. In this introduction we situate our research and present the contributions made by this thesis. This chapter contains five sections:

- Section 1.1 discusses recent developments in AI and the increasing need for labeled data resulting in the so-called *data bottleneck*.
- Section 1.2 reviews some existing alternatives for traditional supervision in general machine learning research motivated by the lack of labeled training data, and provides simple, working definitions of established techniques.
- Section 1.3 then zooms in on several successful applications of efficient supervision applied to NLP.
- We then conclude the introduction by summarizing the contributions made by this thesis in Section 1.4, and present the accompanying papers Section 1.5. Each chapter presents a particular research question in the context of tackling a NLP task from the perspective of applying supervision more efficiently.

1.1 The Data Bottleneck

"Perhaps the most important news of our day is that datasets – not algorithms – might be the key limiting factor to development of human-level

artificial intelligence," Alexander Wissner-Gross argued in a written response to the question raised by technology blog Edge: "What do you consider the most interesting recent scientific news?".[1] Getting labeled training data has become a key development bottleneck for supervised machine learning.

At the dawn of the field of artificial intelligence, two of its founders, Seymour Papert and Marvin Minksy predicted that solving the problem of computer vision would only take a summer [3]. We know now that they were off by at least half a century. One can ask: what took the research community this long? By reviewing the timing of several of the recent most publicized AI advances, he found evidence that suggests an explanation, maybe many major AI breakthroughs have been constrained by the availability of high-quality training datasets, not by algorithmic advances.

Table 1.1 provides an overview of several key AI advances along with enabling algorithms and datasets.[2] Examining these advances, the average elapsed time between the key algorithm proposals and corresponding advances is about ten years, whereas the average elapsed time between key dataset availability and the corresponding advance was less than three years, or about six times faster, suggesting that datasets might have been limiting factors in the advances.

Many of the recent breakthroughs in computer vision and NLP are due to the advent of *deep learning* (neural net architectures with many hidden layers), which allow ML practitioners to get state-of-the-art scores without using hand-engineering features. Whereas building an image classification model ten years ago required advanced knowledge of tools like Sobel operators and Fourier analysis to craft an adequate set of features for a model, deep learning models learn expressive representations inherently from the raw data. Moreover, given the availability of multiple professional-quality open-source machine learning frameworks such as TensorFlow[3] and PyTorch[4], combined with an abundance of available state-of-the-art models, it can be argued that high-quality machine learning models are almost a commoditized resource now.

A caveat with such models is that they tend to rely on massive sets of often hand-labeled training data. Deep learning models are massively more complex than most traditional models: many standard deep learning models today have hundreds of millions of free parameters and thus require more labeled training data. These hand-labeled training sets are expensive and time-consuming to create, taking months or years for large

[1] https://www.edge.org/response-detail/26587
[2] Table based on https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html
[3] https://www.tensorflow.org
[4] http://pytorch.org

	Breakthroughs in AI	Key Dataset		Algorithm
1994	Human level speech recognition	Spoken Wall Street Journal articles and other texts	1986	Hidden Markov Model [4]
1997	IBM Deep Blue defeats Gary Kasparov	700,000 Grandmaster chess games, aka the Extended book	1983	Negascout planning algorithm [5]
2005	Google's Arabic- and Chinese-to-English translation	1.8 trillion tokens from Google Web and News pages (collected in 2005)	1990	Statistical Machine Translation [6]
2011	IBM Watson becomes the world Jeopardy! champion	8.6 million documents from Wikipedia, Wiktionary, Wikiquote and Project Gutenberg	1991	Mixture-of-Experts [7]
2014	Google's GoogLeNet object classification at near-human performance	ImageNet corpus of 1.5 million labeled images and 1,000 object categories [8]	1995	Convolution Neural Networks [9]
2015	Google's Deepmind achieves human parity in playing 29 Atari games by learning general control from video	Arcade Learning Environment datasets for over 50 Atari games	1992	Q-learning algorithm [10]

Table 1.1: Overview of key breakthroughs in machine learning research with the key datasets and algorithms.

benchmark sets, or when domain expertise is required. Afterwards, the labeled datasets often can not be practically repurposed for new objectives. The cost and inflexibility of hand-labeling of such training sets is the key bottleneck to actually deploying machine learning in many cases.

Therefore, in practice today, most large deep learning systems actually use some form of *weak supervision*: noisier, lower-quality, but large-scale training sets constructed via alternative strategies such as using less expensive annotators, scripts, or more creative and higher-level input from domain experts.

1.2 Beyond Traditional Supervision

As discussed in previous section, current supervised models require large quantities of labeled data. A distinction can be made between traditional supervised learning and semi- or weakly supervised methods.

In traditional supervised learning the goal is, given a training set made of pairs (x_i, y_i), to learn a mapping from x to y. Here, the $y_i \in Y$ are called the labels or targets of the examples x_i . There are two families of algorithms for supervised learning. Generative algorithms try to model the joint probability distribution $p(x, y)$ by some unsupervised learning procedure. Discriminative algorithms do not try to characterize items x for a given label y but instead concentrate on estimating $p(y|x)$ directly.

Many traditional lines of research in machine learning are motivated by the appetite of modern machine learning models for labeled training data and propose more efficient techniques. These are divided at a high-level by what information they leverage and how they leverage it. Figure 1.1 provides an overview of different established sources for supervision in machine learning.[4] We briefly describe some of the most prominent techniques.

1.2.1 Semi-Supervised Learning

Semi-supervised learning (SSL) is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision but not necessarily for all examples. Often, this information will be the targets associated with some of the examples. The standard setting in SSL involves the availability of two types of data:

- Labeled data $X_u = x_1, \ldots, x_l$, for which labels $Y_l = y_1, \ldots, y_l$ are provided.

[4] Overview and Figure based on https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html

Figure 1.1: Overview of sources of supervision in machine learning.

- Unlabeled data $X_u = x_{l+1}, \ldots, x_{l+u}$, a much larger set but its labels are not known.

At a high level, assumptions are made about smoothness, low dimensional structure, or distance metrics to leverage the unlabeled data (either as part of a generative model, as a regularizer for a discriminative model, or to learn a compact data representation), for a recent survey see [11]. Broadly, rather than soliciting more input from experts, the idea in SSL is to leverage domain- and task-agnostic assumptions to exploit the unlabeled data that is often cheaply available in large quantities.

More recent methods use generative adversarial training [12], heuristic transformation models [13], and other generative approaches to effectively help regularize decision boundaries.

1.2.2 Active Learning

In **active learning**, the goal is to make use of labels by experts more efficiently by having them label data points which are estimated to be most valuable to the model (for a recent survey, see [14]). Traditionally, applied to the standard supervised learning setting, this means intelligently selecting new data points to be labeled. Common strategies to select data points are based on the uncertainty of the prediction, the expected model change or the expected error reduction.

1.2.3 Multi-Task Learning

When training models for a single task, labels from related tasks can provide additional signal to do better on the metric which is optimized. By sharing representations between related tasks, models are able to generalize better on the original task. This approach is called **Multi-Task Learning**. Multi-task learning has been used successfully across all applications of machine learning, from speech recognition [15] to computer vision [16] and drug discovery [17].

1.2.4 Transfer Learning

In the **transfer learning** setting, the goal is to take one or more models trained on a different dataset and apply them to our dataset and task; for a recent survey, see [18]. A common transfer learning approach in the deep learning community is to “pre-train" a model on one large dataset, and then “fine-tune" it on the task of interest. Another related line of work is multi-task learning, where several tasks are learned jointly [19]. Some

transfer learning approaches take one or more pre-trained models (potentially with some heuristic conditioning of when they are each applied) and use these to train a new model for the task of interest.

1.2.5 Weak Supervision

The aforementioned machine learning paradigms potentially potentially facilitate training without having to rely on expensive annotators for many additional training labels. An alternative option is to provide supervision at a higher-level which is arguably faster and easier but otherwise less precise. This is the key motivation for weak supervision approaches.

In the **weak supervision** setting, the objective is the same as in the supervised setting, however, instead of a ground-truth labeled training set we have:

- Unlabeled data $X_u = x_1, \ldots, x_N$
- One or more weak supervision sources $p_i(y|x), i = 1, \ldots, M$ provided by an expert, where each source provides a weak label to the data.

Weak labels serve as a way for human supervision to be provided more cheaply and efficiently: either by providing higher-level, less precise supervision (e.g., heuristic rules, expected label distributions), cheaper, lower-quality supervision (e.g., crowdsourcing), or taking advantage of existing resources (e.g., knowledge bases, pre-trained models, labeled text collections). These weak label distributions could thus take one of many forms:

- *Deterministic functions*: The weak label distributions could be deterministic functions. We might have a set of noisy labels for each data point. These could come from crowd workers, be the output of heuristic rules $f_i(x)$, or the result of distant supervision [20], where an external knowledge base is heuristically mapped onto unlabeled data X_u. These could also be the output of other classifiers which only yield maximum a posteriori estimates, or which are combined with the heuristic rules to output discrete labels.
- *Constraints*: One can also consider constraints represented as weak label distributions. For example, a structured prediction setting leads to a wide range of very interesting constraint types, such as physics-based constraints [21] or output constraints on the execution of logical forms [22], which encode various forms of domain expertise cheaper supervision from, e.g., layman annotators.

- *Distributions*: We might also have direct access to a probability distribution. For example, we could have the posterior distributions of one or more weak (i.e., low accuracy or coverage) or biased classifiers, such as classifiers trained on different data distributions as in the transfer learning setting. We could also have one or more user-provided label or feature expectations or measurements [22, 23], i.e., an expected distribution $p_i(y)$ or $p_i(y|f(x))$ (where $f(x)$ is some feature of x) provided by a domain expert as in e.g., in [24]

- *Invariances*: Finally, given a small set of labeled data, we can express functional invariances as weak label distributions, e.g., extend the coverage of the labeled distribution to all transformations of $t(x)$ or x, and set $p_i(y|t(x)) = p_i(y|x)$. Techniques such as data augmentation (see Section 1.3.5) can be seen as a form of weak supervision as well.

1.3 Efficient Supervision for NLP

SSL has a long history in NLP dating back to the 1990s, mostly due to applications in text classification problems [25, 26]. In this section we provide an overview of more recent developments and important uses of SSL and weak supervision for NLP.

1.3.1 Semi-Supervised Learning for NLP

Two prominent SSL methods used in NLP are self-training and co-training.

The term self-training has been used to refer to a variety of schemes for using unlabeled data. Ng and Cardie [27] implement self-training by bagging and majority voting. A committee of classifiers are trained on the labeled examples, then classify the unlabeled examples independently. Only those examples, to which all the classifiers give the same label, are added to the training set and those classifiers are retrained. Self-training has been applied to NER [28], machine translation [29], parsing [30] and text classification [31].

Co-training [25] is another algorithm for learning from labeled and unlabeled data which assumes that each data point can be described by two distinct models of the data. Co-training learns two classifiers, one for each view, Dasguptaet al. [32] show that the classifier trained on one view has low generalization error if it agrees on unlabeled data with the classifier trained on the other view. Co-training has been applied to large-scale document classification [33], word sense disambiguation [34], named entity classification [35], statistical parsing [36] and part-of-speech tagging [37].

Tri-training [38] is another *multi-view* training method. Tri-training leverages the agreement of three independently trained models to reduce the bias of predictions on unlabeled data.

1.3.2 Distant Supervision

One of the most proficient forms of weak supervision in NLP is distant supervision (DS). DS automatically labels its own training data by heuristically aligning facts from a knowledge base with an unlabeled corpus.

The last decade of machine learning-based information extraction research has focused on models that require large amounts of labeled data. However, most real-world information extraction tasks do not have any fully labeled data. Labeling new data to train a reasonably accurate sequence model is not only expensive, it also requires labeling data for each new domain.

Mintz et al. [39] first propose the term distant supervision and are the first to use Freebase as the database to generate training data for a knowledge-base population task. The positive training data is obtained by a simple textual match with facts in the knowledge base, special negative training data is included which is generated from entity pairs that are in none of the considered relations according to the knowledge base. A multi-class logistic classifier is used with lexical and named-entity-tag features, as well as features derived from dependency trees.

Since its introduction, DS has known many applications and extensions, most of which find ways to reduce the noise generated by the procedure [40–44].

1.3.3 Information Extraction using Weak Supervision

An interesting line of research using weak supervision are end-to-end data pipelines for information extraction systems. DeepDive [45] introduced a data management system that enables extraction, integration, and prediction problems in a single system, which allows users to rapidly construct sophisticated end-to-end data pipelines by programming features. It views every piece of data as an imperfect observation which may or may not be correct. It uses these observations and domain knowledge expressed by the user to build a statistical model. DeepDive-based systems are used by users without machine learning expertise in a number of domains from paleobiology to genomics to human trafficking. DeepDive was commercialized as Lattice Data and acquired by Apple in 2016 for $200M.[5]

[5] https://techcrunch.com/2017/05/13/apple-acquires-ai-company-lattice-data-a-specialist-in-unstructured-dark-data/

Research on DeepDive was continued under the name Snorkel [46] which focuses on programming weak supervision instead of feature engineering. Snorkel enables users to train models without hand labeling any training data. Instead, users write labeling functions that express arbitrary heuristics, which can have unknown accuracies and correlations.

Snorkel then de-noises their outputs without access to the ground truth by incorporating an end-to-end implementation of their proposed machine learning paradigm, data programming [47]. By modeling a noisy training set creation process in this way, Snorkel can take potentially low-quality labeling functions from the user, and use these to train high-quality end models. In a user study, subject matter experts build models 2.8× faster than manually labeling data and increase predictive performance by on average 45.5% compared to systems needing seven hours of hand labeling.

1.3.4 Crowdsourcing

Many prominent datasets enabling deep learning for NLP are based on crowdsourcing. Crowdsourcing is a way of generating annotation labels cheaply and is an increasingly utilized source of annotation labels to computational linguists as a source of labeled training data to use in machine learning. Recent prominent datasets include the SNLI corpus for text entailment by Bowman et al. [48] and the SQUAD dataset by Rajpurkar et al. [49] for question answering, which enabled the application of deep learning models which consequently obtained state-of-the-art performance on these tasks. Another example is the Fake News challenge [50] in which the task of fact verification was modelled as stance classification. A dataset was generated with claims and corresponding articles curated and labeled by journalists in the context of the Emergent Project [51].

In a common annotation task, multiple labels are collected for an instance, and are aggregated together into a single aggregated label. While crowdsourcing offers solicitors of information or services nearly unlimited cheap labels, the major challenge lies in aggregating the multiple, noisy contributor inputs to create a consistent corpus. The output of crowdsourcing, especially in the case of micro-tasks and when monetary incentives are involved, often suffers from low quality. Moreover, crowd workers are usually not experts and they are of different age, education and ethnics. A high number of labels is needed to compensate for worker bias, task misunderstanding, lack of interest, incompetence, and malicious intent [52].

1.3.5 Data Augmentation for NLP

Data augmentation aims to create additional training data by producing variations of existing training examples through transformations, which can mirror those encountered in the real world. Data augmentation has a proven track record in computer vision, common augmentation techniques are mirroring, random cropping, shearing, etc. For instance, it has been used very effectively in AlexNet [53] to combat overfitting and in most state-of-the-art models since. In addition, data augmentation makes intuitive sense as it makes the training data more diverse and should thus increase a model's ability to generalize.

In NLP, data augmentation is far less obvious to realize and hence, has seen little success so far. In speech recognition, data is augmented by adding artificial background noise and changing the tone or speed of speech signal [54]. In terms of text, it is not reasonable to augment data using signal transformations as done in image or speech recognition, because the exact order of characters may form rigorous syntactic and semantic meaning. Zhang et al. [55] do data augmentation for training of convolutional neural networks for language understanding by replacing words or phrases with synonyms. Xie et al. [56] replace words with samples from different distributions for language modelling and machine translation. Recent work focuses on creating adversarial examples either by replacing words or characters [57], concatenation [58], or adding adversarial perturbations [59]. An adversarial setup is also used by Li et al. [60] who train a system to produce sequences that are indistinguishable from human-generated dialogue utterances.

Back-translation [61] is a common data augmentation method in machine translation that allows to incorporate monolingual training data. For instance, when training a EN-FR system, monolingual French text is translated to English using an FR-EN system, the synthetic parallel data can then be used for training. Back-translation can also be used for paraphrasing [62]. Paraphrasing has also been used for data augmentation in question answering systems [63]. Another method that is close to the use of paraphrases is generating sentences from a continuous space using a variational autoencoder [64].

1.3.6 Transfer Learning for NLP

In the transfer learning setting, a model trained for a different task is applied to a different task of interest. Transfer learning has had a large impact on computer vision and has greatly lowered the entry threshold for people wanting to apply computer vision algorithms to their own problems.

Computer vision practitioners are no longer required to perform extensive feature-engineering for every new task, but can simply start from a model pre-trained on a large dataset and fine-tune it further with a small number of examples for their particular task at hand.

In NLP, transfer learning mostly focused on initializing the first layer of a neural network architecture using pre-trained word representations or embeddings. Recent approaches [65] add pre-trained language model embeddings, but these still require custom architectures for every task. To unlock the true potential of transfer learning for NLP, models need to be pre-trained and fine-tuned on the target task, akin to fine-tuning ImageNet models. Language modeling, for instance, is a great task for pre-training and could be to NLP what ImageNet classification is to CV [66].

Next to word embeddings, sentence representations have also been pre-trained e demonstrated sentence embeddings outperform state-of-the-art unsupervised and supervised representation learning methods on several downstream NLP tasks that involve understanding sentence semantics while achieving an order of magnitude speedup in training time [67].

1.3.7 Multi-Task Learning for NLP

Multi-task learning (MTL) has become more commonly used in NLP [68–70]. One of the main questions is, which tasks are useful for multi-task learning. Language modelling has been shown to be beneficial for many NLP tasks and can be incorporated in various ways. Most word embeddings are trained using an objective similar to language modelling; languages models have been used to pre-train machine translation and sequence-to-sequence models [71], contextual language model embeddings have also been found useful for many tasks [65]. One of the main questions in this domain is determining which NLP tasks are useful for multi-task learning.

1.4 Research contributions

Now that we have introduced techniques for more efficient supervision in machine learning for NLP, we present the core research topics contributed by this thesis. Each chapter presents a core NLP problem that is tackled using new techniques, requiring less supervision and annotation effort by domain experts. In Table 1.2 we provide an overview of the different NLP tasks at hand and the source of (weak) supervision. One will notice that for none of the applied models or techniques a fully labeled collection by domain experts is put forward.

Table 1.2: Overview of contributions presented in this thesis.

Chapter	Task	Source of Supervision
2	Knowledge base population	Distant supervision and small amounts of feature annotations
3	Topic model evaluation	Small existing datasets
4	Automatic keyphrase extraction	Single opinions by layman annotators
5	Sequence-to-sequence modeling	Crowd sourced data from a gamified platform and noisy parallel text

Chapter 2 – Weak Supervision for Knowledge Base Population

Knowledge Base Population (KBP) is the process of populating a knowledge base (KB) from unstructured and/or structured input, e.g., , filling a relational database with facts extracted from text, tables, or even maps and figures. Because KBs are often manually constructed, they tend to be incomplete. KBP systems are crucial to keep KBs up-to-date by extracting information from unstructured data such as webtext. An important component of a KBP system is the relation extractor which detects relations occurring between entities. Training such relation extractors for the purpose of automated knowledge base population requires the availability of sufficient training data. Fortunately, the amount of manual labeling can be significantly reduced by applying distant supervision, which generates training data by aligning large text corpora with available knowledge bases. Yet, this typically results in a highly noisy training set, where many training sentences do not express the intended relation. This chapter presents a method for training relation extractors in knowledge base population systems at very low manual label effort. We use *low dimensional representations of shortest dependency paths between entities of interest to bootstrap classifiers for relation extraction*. We show that at only minutes of labeling per relation we are able to match or improve on accuracy of fully supervised relation extractors. By applying this technique in a participation in the Knowledge Base Population shared task, we achieved top-ranking submissions. These KBP systems are described in more detail in Appendix A.

Chapter 3 – Topic Model Evaluation

In chapter 3 we present an efficient method for evaluation of topic models by making use of existing categorized text collections. While state-of-the-art unsupervised topic models lead to reasonable statistical models of doc-

uments, they offer no guarantee of creating topics that are interpretable by humans and require a manual evaluation of interpretability of the output. Existing methods evaluate statistical goodness-of-fit, offer no guarantee of interpretability. Alternatively an evaluation would require full supervision which can be costly for large topic models. We use small available labeled text collections to provide us with reference topics and present *a new measure for topic quality based on alignment*. Our proposed measure shows a higher correlation with human evaluation than existing unsupervised measures.

Chapter 4 – Creation and Evaluation of Large-scale Collections

Automatic keyphrase extraction (AKE) is the task of automatically extracting the most important and topical phrases of a document [72]. Keyphrases are meant to cover all document topics and capture the complete content of a document in but a handful of phrases. Applications of keyphrases are rich and diverse, ranging from document summarization [73] to clustering [74], contextual advertisement [75], or simply to enhance navigation through large corpora. While several Automatic Keyphrase Extraction (AKE) techniques have been developed and analyzed, there is little consensus on the definition of the task and a lack of overview of the effectiveness of different techniques. Proper evaluation of keyphrase extraction requires large test collections with multiple opinions, which were before, not available for research at the start of our research. In Chapter 4, we present *a set of test collections derived from various sources with multiple, noisy annotations* (which we also refer to as *opinions* in the chapter) for each document, systematically evaluate keyphrase extraction using several supervised and unsupervised AKE techniques, and experimentally analyze the effects of disagreement between annotators on AKE evaluation. Next to *benchmarking existing techniques we study the influence of aggregating multiple annotations in the training data* on the performance metrics. We conclude this chapter by rephrasing the supervised keyphrase extraction problem as positive unlabeled learning in which *a binary classifier is learned in a semi-supervised way from only positive keyphrases and unlabeled candidate phrases*. We show that using only a single opinion per document we are able to achieve scores similar to models trained using multiple opinions per document.

A disadvantage of supervised approaches is that they require a lot of training data and show bias towards the domain on which they are trained, undermining their ability to generalize well to new domains. Unsupervised approaches are a viable alternative in this regard. In Appendix B

we make two focused contributions to the area of unsupervised keyphrase extraction by studying the use of to topic models in graph-based word ranking models.

Chapter 5 – Sequence-To-Sequence Models on Noisy Training Data

In the final research chapter, we present two applications of sequence-to-sequence models trained on noisy or poorly aligned training data. Sequence-to-Sequence models [76, 77] are one of the most successful applications of neural network architectures to natural language processing, providing state-of-the-art results for machine translation. However, these architectures, mostly relying on recurrent neural networks, are heavily parameterized and require large amounts of high-quality training data. In this chapter we study two applications where only noisy or low quality training data is available. In a first setting we present *the novel task of automated lyric annotation and an accompanying dataset providing explanations to lyrics and poetic text*. These models generate explanations to poetic text. The created dataset is one of the largest of its kind and stimulates research in text normalization, metaphor processing and paraphrasing.

In the second part of this chapter, we *extend sequence-to-sequence models with the possibility to control the characteristics or style* of the generated output, via attention that is generated a priori (before decoding) from a latent code vector. After training an initial attention-based sequence-to-sequence model, we use a variational auto-encoder conditioned on representations of input sequences and a latent code vector space to generate attention matrices. By sampling the code vector from specific regions of this latent space during decoding and imposing prior attention generated from it in the sequence-to-sequence model, output can be steered towards having certain attributes. This is demonstrated for the task of sentence simplification, where the latent code vector allows control over output length and lexical simplification, and enables fine-tuning to optimize for different evaluation metrics.

1.5 Publications

The research results obtained during this PhD research have been published in scientific journals and presented at a series of international conferences and workshops. The following list provides an overview of these publications.

1.5.1 Publications in international journals (listed in the Science Citation Index[6])

I **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Knowledge base population using semantic label propagation*. Published in Knowledge Based Systems, 108:79–91, 2016.

II **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Creation and evaluation of large keyphrase extraction collections with multiple opinions*. Published in Language Resources and Evaluation, online, 2017.

1.5.2 Publications in international conferences (listed in the Science Citation Index[7])

III **L. Sterckx**, T. Demeester, J. Deleu, L. Mertens and C. Develder, *Assessing quality of unsupervised topics in song lyrics* . Published in Lecture notes in computer science, Presented at the European Conference on Information Retrieval. 8416:547–552, Amsterdam, Netherlands, 2014.

IV **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *When topic models disagree: keyphrase extraction with mulitple topic models* . Published in WWW'15 companion : Proceedings of the 24th International Conference on World Wide Web, p. 123–124, Florence, Italy, 2015.

V **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Topical word importance for fast keyphrase extraction*. Published in WWW'15 companion : Proceedings of the 24th International Conference on World Wide Web, p. 121–122 Florence, Italy, 2015.

VI B. Vandersmissen, **L. Sterckx**, T. Demeester, A. Jalalvand, W. De Neve and R. Van de Walle, *An automated end-to-end pipeline for fine-grained video annotation using deep neural networks*. Published in ICMR'16: Proceedings of the 2016 ACM International Conference on Multimedia Retrieval, p409–412, New York, New York, USA, 2016.

[6]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

[7]The publications listed are recognized as 'P1 publications', according to the following definition used by Ghent University: P1 publications are proceedings listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

1.5.3 Publications in other international conferences

VII **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Ghent University-IBCN participation in TAC-KBP 2014 slot filling and cold start tasks* Published in 7th Text Analysis Conference, Proceedings. p.1-10, Gaithersburg (MD), USA, 2014

VIII **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Using active learning and semantic clustering for noise reduction in distant supervision* Published in Fourth Workshop on Automated Base Construction at NIPS2014, Proceedings. p.1-6, Montreal, Canada, 2014.

IX **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Ghent University-IBCN participation in TAC-KBP 2015 slot filling and cold start tasks* Published in 8th Text Analysis Conference, Proceedings. p.1-10,Gaithersburg (MD), USA, 2015.

X **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Supervised keyphrase extraction as positive unlabeled learning*. Published in the Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, p. 1924–1929, Austin (Texas), USA, 2016.

XI **L. Sterckx**, T. Demeester, J. Deleu and C. Develder, *Break it down for me : a study in automated lyric annotation*. Published in the Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, p. 2074–2080, Copenhagen, Denmark, 2017.

XII **L. Sterckx**, J. Deleu, C. Develder and T. Demeester, *Prior Attention for Style-aware Sequence-to-Sequence Models*. Submitted to Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018.

References

[1] J. Goethe, E. Stopp, and P. Hutchinson. *Maxims and Reflections*. Classics Series. Penguin Books Limited, 1998. Available from: https://books.google.be/books?id=ZBi2WV01yIwC.

[2] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. *Supervised machine learning: A review of classification techniques*. Emerging artificial intelligence applications in computer engineering, 160:3–24, 2007.

[3] K. Richardson. *An anthropology of robots and AI: annihilation anxiety and machines*, volume 20. Routledge, 2015.

[4] L. Rabiner and B. Juang. *An introduction to hidden Markov models*. IEEE ASSP Magazine, 3(1):4–16, 1986.

[5] A. Reinefeld. *An improvement of the Scout tree-search algorithm*. ICCA Journal, 6(4):4–14, 1983.

[6] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. *A statistical approach to machine translation*. Computational linguistics, 16(2):79–85, 1990.

[7] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. *Adaptive Mixtures of Local Experts*. Neural Computation, 3(1):79–87, March 1991. Available from: http://dx.doi.org/10.1162/neco.1991.3.1.79, doi:10.1162/neco.1991.3.1.79.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *Imagenet: A large-scale hierarchical image database*. In Computer Vision and Pattern Recognition, 2009. CVPR 2009., pages 248–255. IEEE, 2009.

[9] Y. LeCun, Y. Bengio, et al. *Convolutional networks for images, speech, and time series*. The handbook of brain theory and neural networks, 3361(10):1995, 1995.

[10] C. J. Watkins and P. Dayan. *Q-learning*. Machine learning, 8(3-4):279–292, 1992.

[11] O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

[12] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. *Improved Techniques for Training GANs*. CoRR, abs/1606.03498, 2016. Available from: http://arxiv.org/abs/1606.03498, arXiv:1606.03498.

[13] S. Laine and T. Aila. *Temporal Ensembling for Semi-Supervised Learning*. 2017.

[14] B. Settles. *Active learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, 6(1):1–114, 2012.

[15] L. Deng, G. Hinton, and B. Kingsbury. *New types of deep neural network learning for speech recognition and related applications: An overview*. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 8599–8603. IEEE, 2013.

[16] R. Girshick. *Fast R-CNN*. In Computer Vision (ICCV), 2015 IEEE International Conference on, pages 1440–1448. IEEE, 2015.

[17] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande. *Massively multitask networks for drug discovery*. arXiv preprint arXiv:1502.02072, 2015.

[18] S. J. Pan and Q. Yang. *A survey on transfer learning*. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.

[19] S. Ruder. *An overview of multi-task learning in deep neural networks*. arXiv preprint arXiv:1706.05098, 2017.

[20] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. *Distant supervision for relation extraction without labeled data*. (2005), 2008.

[21] R. Stewart and S. Ermon. *Label-Free Supervision of Neural Networks with Physics and Domain Knowledge.* In AAAI, pages 2576–2582, 2017.

[22] J. Clarke, D. Goldwasser, M.-W. Chang, and D. Roth. *Driving Semantic Parsing from the World's Response*. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL '10, pages 18–27, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. Available from: http://dl.acm.org/citation.cfm?id=1870568.1870571.

[23] K. Guu, P. Pasupat, E. Z. Liu, and P. Liang. *From language to programs: Bridging reinforcement learning and maximum marginal likelihood*. arXiv preprint arXiv:1704.07926, 2017.

[24] G. Druck, B. Settles, and A. McCallum. *Active learning by labeling features*. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, pages 81–90. Association for Computational Linguistics, 2009.

[25] A. Blum and T. Mitchell. *Combining Labeled and Unlabeled Data with Co-training*. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT' 98, pages 92–100, New York, NY, USA, 1998. ACM. Available from: http://doi.acm.org/10.1145/279943.279962, doi:10.1145/279943.279962.

[26] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. *Text classification from labeled and unlabeled documents using EM*. Machine learning, 39(2-3):103–134, 2000.

[27] V. Ng and C. Cardie. *Bootstrapping coreference classifiers with multiple machine learning algorithms*. In Conference on Empirical methods in natural language processing, EMNLP 2003, pages 113–120. Association for Computational Linguistics, 2003.

[28] B. Rosenfeld and R. Feldman. *Using Corpus Statistics on Entities to Improve Semi-supervised Relation Extraction from the Web*. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 600–607. Association for Computational Linguistics, 2007. Available from: http://www.aclweb.org/anthology/P07-1076.

[29] N. Ueffing, G. Haffari, and A. Sarkar. *Transductive learning for statistical machine translation*. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 25–32, 2007.

[30] D. McClosky, E. Charniak, and M. Johnson. *Effective self-training for parsing*. In Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics, pages 152–159. Association for Computational Linguistics, 2006.

[31] T. Joachims. *Transductive inference for text classification using support vector machines*. In ICML, volume 99, pages 200–209, 1999.

[32] S. Dasgupta and P. M. Long. *Performance guarantees for hierarchical clustering*. Journal of Computer and System Sciences, 70(4):555–569, 2005.

[33] S.-B. Park and B.-T. Zhang. *Large scale unstructured document classification using unlabeled data and syntactic information*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 88–99. Springer, 2003.

[34] D. Yarowsky. *Unsupervised word sense disambiguation rivaling supervised methods*. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pages 189–196. Association for Computational Linguistics, 1995.

[35] M. Collins and Y. Singer. *Unsupervised models for named entity classification*. In Proc. Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

[36] A. Sarkar. *Applying co-training methods to statistical parsing*. In Proceedings of the 2nd meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, pages 1–8. Association for Computational Linguistics, 2001.

[37] S. Clark, J. R. Curran, and M. Osborne. *Bootstrapping POS taggers using unlabelled data*. In Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003-Volume 4, pages 49–55. Association for Computational Linguistics, 2003.

[38] Z.-H. Zhou and M. Li. *Tri-training: Exploiting unlabeled data using three classifiers*. IEEE Transactions on knowledge and Data Engineering, 17(11):1529–1541, 2005.

[39] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. *Distant supervision for relation extraction without labeled data*. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, pages 1003–1011. Association for Computational Linguistics, 2009.

[40] I. Augenstein, A. Vlachos, and D. Maynard. *Extracting relations between non-standard entities using distant supervision and imitation learning*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 747–757, 2015.

[41] S. Riedel, L. Yao, and A. McCallum. *Modeling relations and their mentions without labeled text*. In Machine Learning and Knowledge Discovery in Databases, pages 148–163. Springer Berlin Heidelberg, 2010.

[42] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. *Knowledge-based weak supervision for information extraction of overlapping relations*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 541–550. Association for Computational Linguistics, 2011.

[43] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. *Multi-instance multi-label learning for relation extraction*. In Proceedings of the

2012 joint conference on empirical methods in natural language processing and computational natural language learning, pages 455–465. Association for Computational Linguistics, 2012.

[44] S. Takamatsu, I. Sato, and H. Nakagawa. *Reducing wrong labels in distant supervision for relation extraction*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pages 721–729. Association for Computational Linguistics, 2012.

[45] C. Zhang, C. Ré, M. Cafarella, C. De Sa, A. Ratner, J. Shin, F. Wang, and S. Wu. *DeepDive: Declarative Knowledge Base Construction*. Communications of the ACM, 60(5):93–102, 2017. doi:10.1145/3060586.

[46] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. *Snorkel: Rapid Training Data Creation with Weak Supervision*. Proc. VLDB Endow., 11(3):269–282, November 2017. Available from: https://doi.org/10.14778/3157794.3157797, doi:10.14778/3157794.3157797.

[47] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré. *Data programming: Creating large training sets, quickly*. In Advances in Neural Information Processing Systems, pages 3567–3575, 2016.

[48] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. *A large annotated corpus for learning natural language inference*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 632–642, 2015.

[49] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. *Squad: 100,000+ questions for machine comprehension of text*. arXiv preprint arXiv:1606.05250, 2016.

[50] D. Pomerleau and D. Rao. *Fake News Challenge* [online]. Available from: http://fakenewschallenge.org/.

[51] W. Ferreira and A. Vlachos. *Emergent: a novel data-set for stance classification*. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies, pages 1163–1168, 2016.

[52] S. Nowak and S. Rüger. *How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation*. In Proceedings of the international conference on Multimedia information retrieval, pages 557–566. ACM, 2010.

[53] A. Krizhevsky, I. Sutskever, and G. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. In NIPS 2012, November 2012.

[54] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. *Deep speech: Scaling up end-to-end speech recognition*. arXiv preprint arXiv:1412.5567, 2014.

[55] X. Zhang and Y. LeCun. *Text understanding from scratch*. arXiv preprint arXiv:1502.01710, 2015.

[56] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng. *Data noising as smoothing in neural network language models*. arXiv preprint arXiv:1703.02573, 2017.

[57] S. Samanta and S. Mehta. *Towards Crafting Text Adversarial Samples*. arXiv preprint arXiv:1707.02812, 2017.

[58] R. Jia and P. Liang. *Adversarial Examples for Evaluating Reading Comprehension Systems*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2021–2031, 2017.

[59] M. Yasunaga, J. Kasai, and D. Radev. *Robust Multilingual Part-of-Speech Tagging via Adversarial Training*. arXiv preprint arXiv:1711.04903, 2017.

[60] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky. *Adversarial learning for neural dialogue generation*. arXiv preprint arXiv:1701.06547, 2017.

[61] R. Sennrich, B. Haddow, and A. Birch. *Improving neural machine translation models with monolingual data*. arXiv preprint arXiv:1511.06709, 2015.

[62] J. Mallinson, R. Sennrich, and M. Lapata. *Paraphrasing Revisited with Neural Machine Translation*. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 881–893, Valencia, Spain, April 2017. Association for Computational Linguistics. Available from: http://www.aclweb.org/anthology/E17-1083.

[63] L. Dong, J. Mallinson, S. Reddy, and M. Lapata. *Learning to paraphrase for question answering*. arXiv preprint arXiv:1708.06022, 2017.

[64] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio. *Generating Sentences from a Continuous Space*. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016, pages

10–21, 2016. Available from: http://aclweb.org/anthology/K/K16/K16-1002.pdf.

[65] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. *Deep contextualized word representations*. arXiv preprint arXiv:1802.05365, 2018.

[66] J. Howard and S. Ruder. *Fine-tuned Language Models for Text Classification*. arXiv preprint arXiv:1801.06146, 2018.

[67] L. Logeswaran and H. Lee. *An efficient framework for learning sentence representations*. arXiv preprint arXiv:1803.02893, 2018.

[68] I. Augenstein, S. Ruder, and A. Søgaard. *Multi-task Learning of Pairwise Sequence Classification Tasks Over Disparate Label Spaces*. arXiv preprint arXiv:1802.09913, 2018.

[69] I. Augenstein and A. Søgaard. *Multi-Task Learning of Keyphrase Boundary Classification*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 341–346, 2017.

[70] M. Rei. *Semi-supervised Multitask Learning for Sequence Labeling*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 2121–2130, 2017.

[71] P. Ramachandran, P. Liu, and Q. Le. *Unsupervised Pretraining for Sequence to Sequence Learning*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 383–391, 2017.

[72] P. D. Turney. *Learning algorithms for keyphrase extraction*. Information retrieval, 2(4):303–336, 2000.

[73] E. D'Avanzo, B. Magnini, and A. Vallin. *Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004*. In Proceedings of the 2004 DUC, 2004.

[74] K. M. Hammouda, D. N. Matute, and M. S. Kamel. *Corephrase: Keyphrase extraction for document clustering*. In International Workshop on Machine Learning and Data Mining in Pattern Recognition, pages 265–274. Springer, 2005.

[75] W.-t. Yih, J. Goodman, and V. R. Carvalho. *Finding advertising keywords on web pages*. In Proceedings of the 15th international conference on World Wide Web, pages 213–222. ACM, 2006.

[76] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112, 2014. Available from: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

[77] D. Bahdanau, K. Cho, and Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. CoRR, abs/1409.0473, 2014. Available from: http://arxiv.org/abs/1409.0473.

2

Weak Supervision for Automatic Knowledge Base Population

This paper presents a method for training relation extractors in knowledge base population systems with very low manual labeling effort. We use low-dimensional representations of shortest dependency paths between entities of interest in order to bootstrap training data. We show that at only minutes of labeling per relation we are able to match or improve on accuracy of fully supervised relation extractors. This technique was developed while participating in the TAC-KBP shared task, and generated top-ranking submissions. Other components of the KBP system are described in more detail in Appendix A. In Section 2.A, we then present a method for cluster-aware active learning to distantly supervised training data.

⋆ ⋆ ⋆

L. Sterckx, T. Demeester, J. Deleu and C. Develder

Appeared in Knowledge Based Systems, online, 2016.

Abstract Training relation extractors for the purpose of automated knowledge base population requires the availability of sufficient training data. The amount of manual labeling can be significantly reduced by applying distant supervision, which generates training data by aligning large text

corpora with existing knowledge bases. This typically results in a highly noisy training set, where many training sentences do not express the intended relation. In this paper, we propose to combine distant supervision with minimal human supervision by annotating features (in particular shortest dependency paths) rather than complete relation instances. Such feature labeling eliminates noise from the initial training set, resulting in a significant increase of precision at the expense of recall. We further improve on this approach by introducing the Semantic Label Propagation (SLP) method, which uses the similarity between low-dimensional representations of candidate training instances to again extend the (filtered) training set in order to increase recall while maintaining high precision. Our strategy is evaluated on an established test collection designed for knowledge base population (KBP) from the TAC KBP English slot filling task. The experimental results show that SLP leads to substantial performance gains when compared to existing approaches while requiring an almost negligible human annotation effort.

2.1 Introduction

In recent years we have seen significant advances in the creation of large-scale knowledge bases (KBs), databases containing millions of facts about persons, organizations, events, products, etc. Examples include Wikipedia-based KBs (e.g., YAGO [1], DBpedia [2], and Freebase [3]), KBs generated from Web documents (e.g., NELL [4], PROSPERA [5]), or open information extraction approaches (e.g., TextRunner [6], PRISMATIC [7]). Other knowledge bases like ConceptNet [8] or SenticNet [9] collect conceptual information conveyed by natural language and make them easily accessible for systems performing tasks like commonsense reasoning and sentiment analysis [10]. Besides the academic projects, several commercial projects were initiated by major corporations like Microsoft (Satori[1]), Google (Knowledge Graph [11]), Facebook[2], Walmart [12] and others. This is driven by a wide variety of applications for which KBs are increasingly found to be essential, e.g., digital assistants, or for enhancing search engine results with semantic search information.

Because KBs are often manually constructed, they tend to be incomplete. For example, 78.5% of *persons* in Freebase have no known *nationality* [13]. To complete a KB we need a knowledge base population (KBP) system that extracts information from various sources of which a large fraction comprises unstructured written text items [11]. A vital component of a

[1]https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing
[2]http://www.insidefacebook.com/2013/01/14/

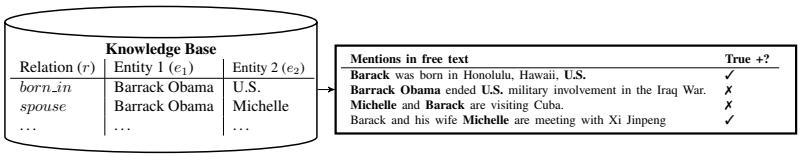


Figure 2.1: Illustration of the distant supervision paradigm and errors

KBP system is a relation extractor to populate a target field of the KB with facts extracted from natural language. Relation extraction (RE) is the task of assigning a semantic relationship between (pairs of) entities in text.

There are two categories of RE systems: (i) *closed*-schema information extraction (IE) systems extract relations from a fixed schema or for a closed set of relations while (ii) *open* domain IE systems extract relations defined by arbitrary phrases between arguments. We focus on the completion of KBs with a fixed schema, i.e., closed IE systems.

Effective approaches for closed schema RE apply some form of supervised or semi-supervised learning [14–19] and generally follow three steps: (i) sentences expressing relations are transformed to a data representation, e.g., vectors are constructed to be used in feature-based methods, (ii) a binary or multi-class classifier is trained from positive and negative instances, and (iii) the model is then applied to new or unseen instances.

Supervised systems are limited by the availability of expensive training data. To counter this problem, the technique of iterative bootstrapping has been proposed [20, 21] in which an initial seed set of known facts is used to learn patterns, which in turn are used to learn new facts and incrementally extend the training set. These bootstrapping approaches suffer from semantic drift and are highly dependent on the initial seed set.

When an existing KB is available, a much larger set of known facts can be used to bootstrap training data, a procedure known as distant supervision (DS). DS automatically labels its own training data by heuristically aligning facts from a KB with an unlabeled corpus. The KB, written as D, can be seen as a collection of relational tables $r(e_1, e_2)$, in which $r \in R$ (R is the set of relation labels), and $< e_1, e_2 >$ is a pair of entities that are known to have relation r. The corpus is written as C.

The intuition underlying DS is that any sentence in C which mentions the same pair of entities (e_1 and e_2) expresses a particular relationship $\hat{r}$ between them, which most likely corresponds to the known fact from the KB, $\hat{r}(e_1, e_2) = r(e_1, e_2)$, and thus forms a positive training example for an extractor of relation r. DS has been successfully applied in many relation extraction tasks [22, 23] as it allows for the creation of large training sets with little or no human effort.

Equally apparent from the above intuition is the danger of finding incorrect examples for the intended relation. The heuristic of accepting each co-occurrence of the entity pair $< e_1, e_2 >$ as a positive training item because of the KB entry $r(e_1, e_2)$ is known to generate noisy training data or false positives [24], i.e., two entities co-occurring in text are not guaranteed to express the same relation as the field in the KB they were generated from. The same goes for the generation of negative examples: training data consisting of facts missing from the KB are not guaranteed to be false since a KB in practice is highly incomplete. An illustration of DS generating noisy training data is shown in Figure 2.1.

Several strategies have been proposed to reduce this noise. The most prominent make use of latent variable models, in which the assumption is made that each known fact is expressed at least once in the corpus [24–26]. These methods are cumbersome to train and are sensitive to initialization parameters of the model.

An active research direction is the combination of DS with partial supervision. Several recent works differ in the way this supervision is chosen and included. Some focus on active learning, selecting training instances to be labeled according to an uncertainty criterion [22, 27], while others focus on annotations of surface patterns and define rules or guidelines in a semi-supervised learning setting [28]. Existing methods for fusion of distant and partial supervision require thousands of annotations and hours of manual labor for minor improvements (4% in F_1 for 23,425 annotations [27] or 2,500 labeled sentences indicating true positives for a 3.9% gain in F_1 [28]). In this work we start from a distantly supervised training set and demonstrate how noise can be reduced, requiring only 5 minutes of annotations per relation, while obtaining significant improvements in precision and recall of the extracted relations.

We define the following research questions:

RQ 1. How can we add supervision more effectively to reduce noise and optimize relation extractors?

RQ 2. Can we combine semi-supervised learning and dimensionality reduction techniques to further enhance the quality of the training data and obtain state-of-the-art results using minimal manual supervision?

With the following contributions, we provide answers to these research questions:

1. In answer to RQ 1, we demonstrate the effectiveness and efficiency of filtering training data based on high-precision trigger patterns. These are obtained by training initial weak classifiers and manually label-

ing a small amount of features chosen according to an active learning criterion.

2. We tackle RQ 2 by proposing a semi-supervised learning technique that allows one to extend an initial set of high-quality training instances with weakly supervised candidate training items by measuring their similarity in a low-dimensional semantic vector space. This technique is called Semantic Label Propagation.

3. We evaluate our methodology on test data from the English Slot Filling (ESF) task of the knowledge base population track at the 2014 Text Analysis Conference (TAC). We compare different methods by using them in an existing KBP system. Our relation extractors attain state-of-the-art effectiveness (a micro averaged F_1 value of 36%) while depending on a very low manual annotation effort (i.e., 5 minutes per relation).

In Section 2.2 we give an overview of existing supervised and semi-supervised RE methods and highlight their remaining shortcomings. Section 2.3 describes our proposed methodology, with some details on the DS starting point (Section 2.3.1), the manual feature annotation approach (Section 2.3.2), and the introduction of the semantic label propagation method (Section 2.3.3). The experimental results are given in Section A.2.4, followed by our conclusions in Section 2.5.

2.2 Related Work

The key idea of our proposed approach is to combine DS with a minimal amount of supervision, i.e., requiring as few (feature) annotations as possible. Thus, our work is to be framed in the context of supervised and semi-supervised relation extraction (RE), and is related to approaches designed to minimize the annotation cost, e.g., active learning. Furthermore, we use compact vector representations carrying semantics, i.e., so-called word embeddings. Below, we therefore briefly summarize related work in the areas of (i) supervised RE, (ii) semi-supervised RE, (iii) evaluations of RE, (iv) active learning and (v) word embeddings.

2.2.1 Supervised Relation Extraction

Supervised RE methods rely on training data in the form of sentences tagged with a label indicating the presence or absence of the considered relation. There are three broad classes of supervised RE: (i) methods based

on manual feature engineering, (ii) kernel based methods, and (iii) convolutional neural nets.

Methods based on feature-engineering [17, 29] extract a rich list of manually designed structural, lexical, syntactic and semantic features to represent the given relation mentions as sparse vectors. These features are cues for the decision whether the relation is present or not. Afterwards a classifier is trained on positive and negative examples. In contrast, *kernel based methods* [19, 30, 31] represent each relation mention as an object such as an augmented token sequence or a parse tree, and use a carefully designed kernel function, e.g., subsequence kernel or a convolution tree kernel, to calculate their similarity with test patterns. These objects are usually augmented with extra features such as semantic information. With the recent success of deep neural networks in natural language processing, Convolutional neural networks (CNNs) have emerged as effective relation extractors [32–34]. CNNs avoid the need for preprocessing and manual feature design by transforming tokens into dense vectors using embeddings of words and extract n-gram based features independent of the position in the sentence.

Supervised approaches all share the need for training data, which is expensive to obtain. Two common methods have emerged for the generation of large quantities of training data, both require an initial set of known instances. When this number is initially small, the technique of *bootstrapping* is used. When a very large number of instances is available from an existing knowledge base, *distant supervision* is the preferred technique. Both are briefly discussed below.

2.2.1.1 Bootstrapping models for Relation Extraction

When a limited set of labeled instances is available, bootstrapping methods have proven to be effective methods to generate high-precision relation patterns [20, 21, 35, 36]. The objective of bootstrapping is to expand an initial 'seed' set of instances with new relationship instances. Documents are scanned for entities from the seed instances and linguistic patterns connecting them are extracted. Patterns are then ranked according to coverage (recall) and low error rate (precision). Using the top scoring patterns, new seed instances are extracted and the cycle is repeated.

An important step in bootstrapping methods is the calculation of similarity between new patterns and the ones in the seed set. This measure decides whether a new pattern is relevant for the relation or not, based on the existing set. Systems use measures based on exact matches [35], cosine-similarity [20] or kernels [36]. A fundamental problem of these methods is semantic drift [37, 38]: bootstrapping, after several iterations,

deviates from the semantics of the seed relationship and extracts unrelated instances which in turn generate faulty patterns. This phenomenon worsens with the number of iterations of the bootstrapping process.

Recently, Batista et al. [39] proposed the use of word embeddings for capturing semantic similarity between patterns. Contexts are modeled using linear combinations of the word embeddings and similarity is measured in the resulting vector space. This approach has shown to reduce semantic drift compared to previous similarity measures.

2.2.1.2 Distant Supervision

Distant supervision (DS) was first proposed in [40], where labeled data was generated by aligning instances from the Yeast Protein Database into research articles to train an extractor. This approach was later applied for training of relation extractors between entities [13] and jointly training the named entity classifier and the relation extractor [41].

Automatically gathering training data with DS is governed by the assumption that *all sentences* containing both entities engaged in a reference instance of a particular relation, represent that relation. Many methods have been proposed to reduce the noise in training sets from DS. In a series of works the labels of DS data are seen as latent variables. Riedel et al. [24] relaxed the strong *all sentences*-assumption to an *at-least-one-sentence*-assumption, creating a multi-instance learner. Hoffman et al. [42] modified this model by allowing entity pairs to express multiple relations, resulting in a multi-instance multi-label setting (MIML-RE). Surdeanu et al. [26] further extended this approach and included a secondary classifier, which jointly modeled all the sentences in texts and all labels in knowledge bases for a given entity pair.

Other methods apply heuristics [43], model the training data as a generative process [44, 45] or use a low-rank representation of the feature-label matrix to exploit the underlying semantic correlated information.

2.2.2 Semi-supervised Relation Extraction

Semi-supervised Learning is situated between supervised and unsupervised learning. In addition to unlabeled data, algorithms are provided with some supervised information. The training data comprises labeled instances $X_l = (x_1 \dots x_l)$ for which labels $Y_l = (y_1 \dots y_l)$ are provided, and typically a large set of unlabeled ones $X_u = (x_1 \dots x_u)$.

Semi-supervised techniques have been applied to RE on multiple occasions. Chen et al. [46] apply label propagation by representing labeled and unlabeled examples as nodes and their similarities as the weights of edges

in a graph. In the classification process, the labels of unlabeled examples are then propagated from the labeled to unlabeled instances according to similarity. Experimental results demonstrate that this graph-based algorithm can outperform SVM in terms of $F_1$when very few labeled examples are available. Sun et al. [18] show that several different word cluster-based features trained on large corpora can compensate for the sparsity of lexical features and thus improve the RE effectiveness.

Zhang et al. [47] compare DS and complete supervision as training resources but do not attempt to fuse them. They observe that DS systems are often recall gated: to improve DS quality, large input collections are needed. They also report modest improvements by adding crowd-sourced yes/no votes to the training instances. Training instances were selected at random as labeling using active learning criteria did not affect performance significantly.

Angeli et al. [27] show that providing a relatively small number of mention-level annotations can improve the accuracy of MIML-RE. They introduce an active learning criterion for the selection of instances incorporating both the uncertainty and the representativeness, and show that a sampling criterion which incorporates not only disagreement but also representativeness in selecting examples to annotate, outperforms existing baselines for active learning.

The MIML-RE model of Surdeanu et al. [26] marginally outperforms the Mintz++ baseline using solely DS: initialization of the latent variables using labeled data is needed for larger improvements. For this, a total of $10,000$ instances were labeled, resulting in a 3% increase on the micro-F_1.

Guided DS as proposed by Pershina et al. [28] incorporates labeled patterns and trigger words to guide MIML-RE during training. They make use of a labeled dataset from TAC KBP to extract training guidelines, which are intended to generalize across many examples.

2.2.3 TAC KBP English Slot Filling

The knowledge base population (KBP) shared task is part of the NIST Text Analysis Conference and aims to evaluate different approaches for discovering facts about entities and expansion of knowledge bases. A selection of entities is distributed among participants for which missing facts need to be extracted from a given large collection of news articles and internet fora. Important components of these systems are query expansion, entity linking and relation extractors. Over the years DS has become a regular feature of effective systems [22, 48]. Other approaches use handcoded rules or are based on question answering systems [48]. The top

performing 2014 KBP ESF system [49] uses DS, the manual labeling of 100,000 features, and is built on DeepDive, a database system allowing users to rapidly construct sophisticated end-to-end knowledge base population techniques [50]. After initial DS, features are manually labeled and only pairs associated with labeled features are used as positive examples. This approach has proven to be very effective but further investigation is needed to reduce the amount of feature labeling. Here, we show how we can strongly reduce this effort while maintaining high precision.

2.2.4 Active Learning and Feature Labeling

Active learning is used to reduce the amount of supervision required for effective learning. The most popular form of active learning is based on iteratively requiring manual labels for the most informative instances, an approach called uncertainty sampling. In relation extraction, typical approaches include query-by-committee [27, 51] and cluster-based sampling [52]. While the focus in RE has been on labeling relation instances, alternative methods have been proposed in other tasks in which features (e.g., patterns, or the occurrence of terms) are labeled as opposed to instances [53, 54], resulting in a higher performance using less supervision.

Getting positive examples for certain relations can be hard, especially when training data is weakly supervised. Standard uncertainty sampling is ineffective in this case: it is likely that a feature or instance has a low certainty score because it does not carry much discriminative information about the classes. Assigning labels to the most certain features has much greater impact on the classifier and can remove the principle sources of noise. This approach has been coined as feature certainty [54], and we show that this approach is especially effective in DS for features that generalize across many training instances.

2.2.5 Distributional Semantics

The Distributional Hypothesis [55] states that words that tend to occur in similar contexts are likely to have similar meanings. Representations of words as dense, low-dimensional vectors (as opposed to the standard one-hot vectors), called word embeddings, exploit this hypothesis and are trained from large amounts of unlabeled text. Representations for words will be similar to those of related words, allowing the model to generalize better to unseen events. The resulting vector space is also called a *vector model of meaning* [56]. Common techniques for generating very dense, short vectors use dimensionality reduction techniques (e.g., singular value decomposition) or neural nets to create so-called word embeddings. Word

embeddings have proven to be beneficial for many natural language processing tasks including POS-tagging, machine translation and semantic role labeling. Two prominent methods for the embedding of words are *Skip-Gram-with-Negative-Sampling* implemented in *Word2Vec* [57], and *GloVe* [58].

While much research has been directed at ways of constructing distributional representations of individual words, for example co-occurrence based representations and word embeddings, there has been far less consensus regarding the representation of larger constructions such as phrases and sentences from these representations. Blacoe et al. [59] show that, for short phrases, a simple composition like addition or multiplication of the distributional word representations is competitive with more complex supervised models such as recursive neural networks.

2.3 Labeling Strategy for Noise Reduction

In this section we introduce our strategy to combine distantly supervised training data with minimal amounts of supervision. Briefly summarized, we designed our labeling strategy such as to *minimize the amount of false positive instances or noise while maintaining the diversity of relation expressions generated by DS*.

We perform a highly selective form of noise reduction starting from a fully distantly supervised relation extractor, described in Section 2.3.1, and use the feature weights of this initial extractor to guide manual supervision in the feature space. Various questions arise from this. When do we overconstrain the original training set generated by DS? What is the trade-off between the application of DS with highly diverse labeled instances, and the constraining approach of labeling features, with a highly accurate yet restricted set of training data? This is discussed in detail in Sections 2.3.2 and 2.3.3.

Our approach is depicted in Figure 2.2, and comprises the following steps:

(1) An existing KB is used to generate distantly supervised training instances by matching its facts with sentences from a large text corpus. We discuss the characteristics of this weakly labeled training set as well as the features extracted from each sentence (see Section 2.3.1).

(2) An initial relation extractor is trained using the noisy training data generated in Step (1).

(3) Confident positive features learned by this initial classifier are pre-

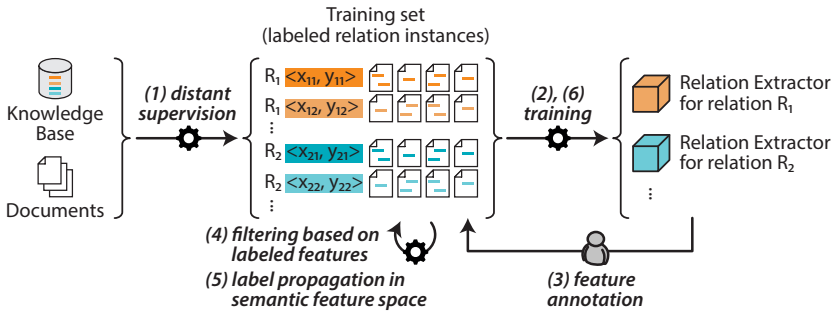


Figure 2.2: Workflow Overview. Note that only Step (3) involves human annotations.

sented to an annotator with knowledge of the semantics of the relation and labeled as true positive or false positive.

(4) The collection of training instances is filtered according to the labeled features and a second classifier is trained. This framework, in which we combine supervision and DS, is explained in Section 2.3.2.

(5) In a *semi-supervised* step, the filtered distantly supervised training data is added to training data by propagating labels from labeled features to distantly supervised instances based on similarity in a semantic vector space of reduced dimension. The technique is presented in Section 2.3.3 as *Semantic Label Propagation*.

(6) A final relation extractor is trained on the augmented training set. We evaluate and discuss results of the proposed techniques in Section A.2.4.

2.3.1 Distantly Supervised Training Data

The English Gigaword corpus [60] is used as unlabeled text collection to generate relation mentions. The corpus consists of 1.8 million news articles published between January 1987 and June 2007. Articles are first preprocessed using different components of the Stanford CoreNLP toolkit [61], including sentence segmentation, tokenizing, POS-tagging, named entity recognition, and clustering of noun phrases which refer to the same entity.

As KB we use a snapshot of Freebase (now Wikidata) from May 2013. The relation schema of Freebase is mapped to that used for evaluation, the NIST TAC KBP ESF Task, which defines 41 relations, including 25 relations with a person as subject entity and 16 with organizations as sub-

ject. 26 relations require objects or fillers that are themselves named entities (e.g., Scranton as place of birth of Joe Biden), whereas others require string-values (e.g., profession (senator, teacher,...), cause of death (cancer, car accident,...)).

We perform weak entity linking between Freebase entities and textual mentions using surface string matching, for which an exact between the name in Freebase and the named entity in the text is needed. We reduce the effect of faulty entity links by thresholding the amount of training data per subject entity [62]. Most frequently occurring entities from the training data (e.g., John Smith, Robert Johnson, ...) are often most ambiguous, hard to link to a KB and thus result in noisy training data. Thresholding the amount of training data per entity also prevents the classifier from overfitting on several, popular entities. This follows from the observation that training data is initially skewed towards several entities frequently occurring in news articles, like Barack Obama or the United Nations, resulting in over-classifying professions of persons as president or seeing countries as members of the organization.

For each generated pair of mentions, we compute various lexical, syntactic and semantic features. Table 2.1 shows an overview of all the features applied for the relation classification. We use these features in a binary logistic regression classifier. Features are illustrated for an example relation-instance <Ray Young, General Motors> and the sentence *"Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi"*.

For each relation R_i, we generate a set of (noisy) positive examples denoted as R_i^+ and defined as

$$R_i^+ = \{ \ (m_1, m_2) \mid R_i(e_1, e_2) EL(e_1, m_1) EL(e_2, m_2) \ \}$$

with e_1 and e_2 being subject and object entities from the KB and $EL(e_1, m_1)$ being the entity e_1 linked to mention m_1 in the text. As in previous work [29, 42], we impose the constraint that both entity mentions $(m_1, m_2) \in R_i^+$ are contained in the same sentence. To generate negative examples for each relation, we sample instances from co-occurring entities for which the relation is not present in the KB.

We measured the amount of noise, i.e., false positives, in the training set of positive DS instances, for a selection of 15 relations: we manually verified 2,000 randomly chosen instances (that DS found as supposedly positive examples) for each of these relations. Table 2.2 shows the percentage of true positives among these 2,000 instances for each of these relations, which strongly varies among relations, ranging from 10% to 90%.

Table 2.1: Overview of different features used for classification for the sentence "Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi".

Feature	Description	Example Feature Value
Dependency tree	Shortest path connecting the two names in the dependency parsing tree coupled with entity types of the two names	PERSON←appos←officer → prep_of→ ORGANIZATION
	The head word for name one	said
	The head word for name two	officer
	Whether *1dh* is the same as *e2dh*	false
	The dependent word for name one	officer
	The dependent word for name two	nil
Token sequence features	The middle token sequence pattern	, the chief financial officer of
	Number of tokens between the two names	6
	First token in between	,
	Last token in between	of
	Other tokens in between	{the, chief, financial, officer}
	First token before the first name	nil
	Second token before the first name	nil
	First token after the second name	,
	Second token after the second name	said
Entity features	String of name one	Ray_Young
	String of name two	General_Motors
	Conjunction of *e1* and *e2*	Ray_Young–General_Motors
	Entity type of name one	PERSON
	Entity type of name two	ORGANIZATION
	Conjunction of *et1* and *et2*	PERSON–ORGANIZATION
Semantic feature	Title in between	True
Order feature	1 if name one comes before name two; 2 otherwise.	1
Parse Tree	POS-tags on the path connecting the two names	NNP→DT→JJ→JJ →NN→IN→NNP

Table 2.2: Training Data. Fractions of true positives are estimated from the training data by manually labeling a sample of 2,000 instances per relation that DS indicated as positive examples

Relation	**Estimated Fraction of True Positives**	**Positively Labeled SDPs**	**Remaining Training Data after Filtering**	**Initial Number of True Positives**
per:title	85.1%	157	26.2%	369,079
org:top_members_employees	71.7%	236	16.7%	93,900
per:employee_or_member_of	87.8%	256	16.5%	260,785
per:age	62.4%	79	52.2%	58,980
per:origin	85.2%	116	11.9%	1,555,478
per:countries_of_residence	55.6%	65	8.4%	493,064
per:charges	59.4%	122	21.5%	17,639
per:cities_of_residence	11.7%	96	7.4%	370,153
per:cause_of_death	51.9%	97	29.4%	31,386
per:spouse	63.2%	124	12.1%	172,874
per:city_of_death	19.9%	92	5.6%	125,333
org:country_of_headquarters	10.8%	92	13.4%	13,435
per:country_of_death	77.6%	70	16.5%	128,773
org:city_of_headquarters	56.5%	67	42.7%	36,238
org:founded_by	13.3%	85	22.7%	318,991

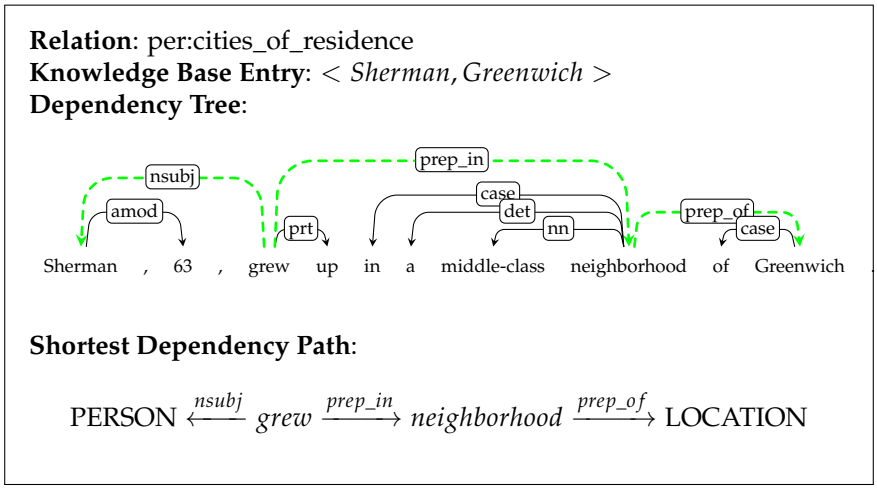


Figure 2.3: Example of a dependency tree feature.

2.3.2 Labeling High Confidence Shortest Dependency Paths

This section describes the manual feature labeling step that allows transforming a full DS training set into a strongly reduced yet highly accurate training set, based on feature labeling. We focus on a particular kind of feature, i.e., a relation's shortest dependency path (SDP). Dependency paths have empirically been proven to be very informative for relation extraction: their capability of capturing information is evidenced by a systematic comparison in effectiveness of different kernel methods [63] or as features in feature-based systems [17]. This was originally proposed by Bunescu et al. [19], who claimed that the relation expressed by a sentence is often captured in the *shortest* path connecting the entities in the dependency graph. Figure 2.3 shows an example of an SDP for a sentence expressing a relation between a person and a city of residence.

As shown in Table 2.2, the fraction of false positive items among all weakly supervised instances can be very large. Labeling features based on the standard active learning approach of uncertainty sampling is ineffective in our case since it is likely that a feature or instance has a low certainty score simply because not much discriminative information about the classes is carried. Annotating many such instances would be a waste of effort. Assigning labels to the most certain features has much greater impact on the classifier and can remove the principal sources of noise. This approach is called feature certainty sampling [54]. It is intuitively an attractive method, as the goal is to reduce the most influential sources of noise as quickly as possible. For example for the relation *founded_by*, there are many persons that founded a company who were also *top_members*,

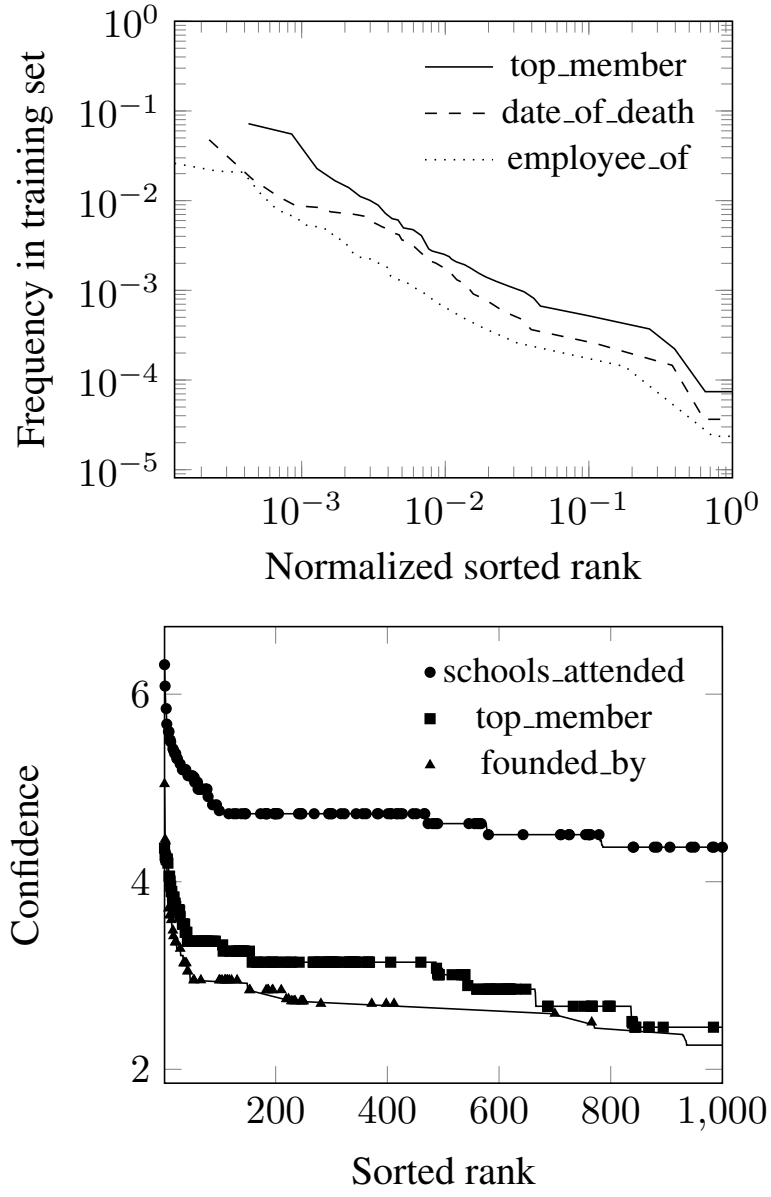


Figure 2.4: Illustration of frequency and confidence of dependency paths for example relations. (a) Occurrence frequency, ranked from highest to lowest, and (b) confidence C of dependency paths (eq. 2.1), ranked from highest to lowest, with indication of true positives.

leading to instances that we wish to remove when cleaning up the training data for the relation *founded_by*. SDPs offer all the information needed to assess the relationship validity of the training instances, are easily labeled, and generalize over a considerable fraction of the training set as opposed to many of the feature-unigrams which remain ambiguous in many cases. We implement the feature certainty idea by ranking SDP features according to the odds that when a particular SDP occurs, it corresponds to a valid relation instance. This corresponds to ranking by the following quantity, which we call the considered SDP's confidence

$$\text{Confidence}(SDP) = \frac{P(+|SDP)}{P(-|SDP)}. \tag{2.1}$$

It can be directly estimated from the original DS training set, based on each SDP feature's (smoothed) occurrence frequencies among the positive and negative distantly supervised instances. In particular, $P(+|SDP)$ indicates the SPD's fraction of occurrences among the positive training data and $P(-|SDP)$) among the negative.

All dependency paths are ranked from most to least confident and the top-k are assigned to a human annotator to select the true positive SDPs. The annotator is asked to select only the patterns which unambiguously express the relation. That is, a pattern is accepted only if the annotator judges it a sufficient condition for that relation. The annotator is provided with several complete sentences containing the dependency path to this cause. When the SDP does not include any verbs, e.g., when entities are both part of the same Noun Phrase like "Microsoft CEO Bill Gates", all words between the subject and object are included and the complete path is added to the filter set. In our experiments, we restrict the time of SDP annotations to a limited effort of 5 minutes for each relation. On average our expert annotator was able to label around 250 SDPs per relation this way. The ease of annotating SDPs becomes apparent when compared with annotating random relation instances, of which they managed to annotate only 100 in the same period of time. Examples of annotated confident patterns are shown in Table 2.3. Section 2.4.3 provides further details on the different annotation methodologies for the experiments.

The motivation behind limiting the annotation time per relation to only a few hundred patterns comes from the following analysis. First of all, a small subset of all different patterns is responsible for the majority of relation instances in the DS training set. In fact, the sparsity of distantly supervised training data becomes apparent when extracting all SDPs for each fact in the KB in one pass over the corpus. Figure 2.4a shows the approximately Zipfian distribution of the frequency of the dependency paths generated by DS in the positively labeled training set for several example relations. The abscis shows the rank of dependency paths for various relations, sorted from most to least frequent, normalized by the total number of paths for the respective relations (to allow visualization on the same

graph). In line with our goal of getting a highly accurate training set with the largest sources of noise removed at a low annotation cost, we focused on capturing those top most frequent patterns. Secondly, we noticed that beyond the first few hundred most confident SDPs, which took around 5 minutes to annotate, further true positives tend to occur less frequently. Annotating many more SDPs would only marginally increase the diversity in the training set, and reduce the gain of . Figure 2.4b illustrates the occurrence of true positive patterns for decreasing confidence scores. For several example relations, the figure shows the true positive patterns as markers on the confidence distribution of the $1,000$ most confident SDPs. We do stress the importance of labeling highly confident SDPs carefully. A false positive SDP can have large effects on the quality of the resulting training data. Therefore, we advise to have multiple annotators annotate the most confident SDPs.

Finally, using the manually selected set of SDPs, the complete training set is filtered by enforcing that one of these SDPs be present in the feature set of the instance. We include all mention pairs associated with that feature as positive examples of the considered relation. The classifier trained on the resulting training set is intuitively of high precision but does not generalize well to unseen phrase constructions. Note that the classifier is quite different from a regular pattern based relation extractor. Although all training instances satisfy at least one of the accepted SDPs, the classifier itself is trained on a set of features including, but not restricted to, these SDPs (see Table 2.1). Still, most of the benefits of DS are lost by having the selection of training instances governed by a limited set of patterns.

The fourth column of Table 2.2 lists the fraction of training data remaining after filtering out all patterns apart from those classified as indicative of the relation at hand. The amount of training data remaining after this filtering step strongly depends on the specific relation, varying from 5% to more than half of the original training set. Yet on the whole, the filtering results in a strong reduction of the purely DS-based training data, often removing much more than the actual fraction of noise (column 2). For example, for the relation *per:employee_or_member_of*, we note only $100\% - 87.8\% = 12.2\%$ false positives, but the manual filtering leads to discarding 83.5% of the DS instances.

The strategy described in the previous paragraphs is related to the *guidelines* strategy from Pershina et al. [28] (without the MIML model) in labeling features, but it differs in some essential aspects. Instead of needing a fully annotated corpus to do so, we rank and label features entirely based on DS. Labeling features based on a fully labeled set ignores the variety of DS and risks being biased towards the smaller set of labeled instances. Also, no active learning criteria were applied when choosing which features to label, making the process even more efficient.

Table 2.3: Examples of top-ranked patterns

Relation	Top SDP	Assessment
top_employees	PER $\xleftarrow{appos}$ *executive* $\xrightarrow{prep\_of}$ ORG	✓
	PER $\xleftarrow{appos}$ *chairman* $\xrightarrow{appos}$ ORG	✓
	ORG $\xleftarrow{nn}$ *founder* $\xrightarrow{prep\_of}$ PER	✗
children	PER-2 $\xleftarrow{appos}$ *son* $\xrightarrow{prep\_of}$ PER-1	✓
	PER-1 $\xleftarrow{appos}$ *father* $\xrightarrow{prep\_of}$ PER-2	✓
	PER-2 $\xleftarrow{nn}$ *grandson* $\xrightarrow{prep\_of}$ PER-1	✗
city_of_birth	PER $\xleftarrow{rcmod}$ *born* $\xrightarrow{prep\_in}$ LOC	✓
	PER $\xleftarrow{nsubj}$ *mayor* $\xrightarrow{prep\_of}$ LOC	✗
	PER $\xleftarrow{appos}$ *historian* $\xrightarrow{prep\_from}$ LOC	✗
schools_attended	PER $\xleftarrow{nsubj}$ *graduated* $\xrightarrow{prep\_from}$ ORG	✓
	PER $\xleftarrow{dep}$ *student* $\xrightarrow{prep\_at}$ ORG	✓
	PER $\xleftarrow{appos}$ *teacher* $\xrightarrow{prep\_at}$ ORG	✗
(org:)parents	ORG-2 $\xleftarrow{appos}$ *subsidiary* $\xrightarrow{prep\_of}$ ORG-1	✓
	ORG-1 $\xleftarrow{appos}$ *division* $\xrightarrow{prep\_of}$ ORG-2	✓
	ORG-2 $\xleftarrow{prep\_to}$ *shareholder* $\xrightarrow{dep}$ ORG-1	✗

2.3.3 Noise Reduction using Semantic Label Propagation

If we strictly follow the approach proposed in Section 2.3.2 and only retain DS training instances that satisfy a positively labeled SDP, an important advantage of DS is lost, namely its potential of reaching high recall. If we limit the feature annotation effort, we risk losing highly valuable SDPs. To counteract this effect, we introduce a second (re)labeling stage, adopting a semi-supervised learning (SSL) strategy to expand the training set. This is done by again adding some instances from the set of previously discarded DS instances with SDPs not matching any of the manually labeled patterns. We rely on the basic SSL approach of self-training by propagating labels from known instances to the nearest neighboring unlabeled instances. This method requires a method of determining the distance between labeled and unlabeled instances. Dangers of self-training include the failure to expand beyond the initial training data or the introduction of errors into the labeled data. In order to avoid an overly strong focus on the filtered training data, we use low-dimensional vector representations of words, also called word embeddings.

Word embeddings allow for a relaxed semantic matching between the

labeled seed patterns and the remaining weakly labeled patterns. As shown by Sterckx et al. [52], representing small phrases by summing each individual word's embedding leads to semantic representations of small phrases that are meaningful for the goal of relation extraction. We represent each relation instance by a single vector by first removing stop-words and averaging the embeddings of the words on the dependency path. For example, consider the sentence:

Geagea on Friday for the first time addressed the court judging him for **murder** charges.

which has the following SDP,

$$\text{PER} \xleftarrow{nsubj} addressed \xrightarrow{dobj} court \xrightarrow{vmod} judging \xrightarrow{prep\_for} charges \xrightarrow{nn} \text{Criminal\_Charge}$$

Its low-dimensional representation $\mathbf{C}$ is hence generated as

$$\mathbf{C} = \frac{E(\text{“addressed”}) + E(\text{“court”}) + E(\text{“judging”}) + E(\text{“charges”})}{4}, \tag{2.2}$$

with $E(x)$ the word embedding of word x. The similarity between a labeled pattern $\mathbf{C}_t$ and a weakly labeled pattern $\mathbf{C}_{DS}$ is then measured using cosine similarity between the vector representations.

$$Sim(\mathbf{C}_t, \mathbf{C}_{DS}) = \frac{\mathbf{C}_t.\mathbf{C}_{DS}}{|\mathbf{C}_t|.|\mathbf{C}_{DS}|} \tag{2.3}$$

In the special case that no verbs occur between two entities, all the words between the two entities are used to build the representations for the context vector.

Using these low-dimensional continuous representations of patterns, we can calculate similarity with longer, less frequently occurring patterns in the training data and the patterns from the initial seed set which are the most frequently occurring ones. We can now increase recall by adding similar but less frequent patterns. More specifically, we calculate the similarity of the average vector of the labeled patterns (as in the Rocchio classifier type of self-training) with each of the remaining patterns in the DS set and extend the training data with the patterns that have a sufficiently high similarity with the labeled ones. We call this technique *Semantic Label Propagation*.

2.4 Experimental Results

2.4.1 Testing Methodology

We evaluate the relation extractors in the context of a Knowledge Base Population system [62, 64] using the NIST TAC KBP English Slot Filling (ESF)

Evaluation from 2012 to 2014. We choose this evaluation because of the diversity and difficulty of entities in the queries. In the end-to-end ESF framework, the input to the system is a given entity (the 'query'), a set of relations, and a collection of articles. The output is a set of slot fillers, where each slot filler is a triple consisting of two entities (including the query entity) and a relation predicted to hold among these entities.

2.4.2 Knowledge Base Population System

Systems participating in the TAC KBP ESF need to handle each task of filling missing slots in a KB. Participants are only provided with one surface-text occurrence of each query entity in a large collection of text provided by the organizers. This means that an information retrieval component is needed to provide the relation extractor with sentences containing candidate fillers. Our system performs query expansion using Freebase aliases and Wikipedia pages. Each document containing one of the aliases is parsed and named entities are automatically detected. Persons, organizations, and locations are recognized, and locations are further categorized as cities, states, or countries. Non-entity fillers like titles or charges are tagged using lists and table-lookups. For further details of the KBP system we refer to Appendix A.

2.4.3 Methodologies for Supervision

In this section we detail the different procedures for human supervision. Supervision is obtained in two forms: by labeling shortest dependency paths (SDPs) and by labeling single training instances indicated as positive by DS, as either true positives or as false positives (noise). After a background corpus is linked with a knowledge base, phrases containing facts are stored in a database for further feature extraction, post processing, and calculation of feature confidence values. Our annotators for the labeling of single training instances were undergraduate students from different backgrounds with little or no experience in machine learning or natural language processing. First, they were briefed on the semantics of the relation to be extracted using the official TAC KBP guidelines. They were then presented with training instances, i.e., phrases from the database. Each instance was shown with entity and subject highlighted and colored. The average time needed to annotate a batch of 2,000 instances was three hours, corresponding to about 5 seconds per instance, including the time needed to read and judge the sentence. As this procedure was relatively expensive (annotators were paid \$15 per hour), only the 15 most frequent relations, strongly influencing the optimal micro-F_1score shown in Table 2.2, were selected. Other relations received between 200 and 1,000 annotations each. In contrast, the time for annotation of the SDPs was limited to merely 5 minutes per relation, during which, on average, 200 SDPs were judged.

SDPs were presented in a spreadsheet as a list, and true positives were labeled using a simple checkbox. All SDP annotations were done by a single expert annotator. To measure the degree of expertise needed for these annotations, we also assigned a novice annotator (student) with the same task. We measured annotator agreement and time needed for a selection of the relations. For this experiment the student was explained the meaning of dependency paths and the aim of choosing valid SDPs. Several lists of SDPs that the expert was able to label in 5 minutes were presented to the student. For the first two relations the student needed more than 10 minutes to label, but for the subsequent relations, annotation time dropped to 5 minutes per relation, equivalent to the time needed by an expert annotator. We measured inter annotator agreement using Cohen's kappa coefficient κ. Inter-annotator agreement between student and expert was initially moderate ($\kappa = 0.65$) and increased after the student completed lists of SDPs for two relations (κ varies between 0.85 and 0.95), indicating a very good agreement.

2.4.4 Pattern-based Restriction vs. Similarity-based Extension

As Table 2.2 shows, applying the manually annotated features as described in Section 2.3.2 often leads to a drastic reduction of training instances, compared to the original distantly labeled training set. Using similarity metrics described in Section 2.3.3, we again add weakly supervised training data to the filtered data. An important question is therefore how to optimally combine initial reduction with subsequent expanding of the training instances. On the one hand, one would expect a high-precision-low-recall effect in the extreme case of adding no similar patterns, and a low-precision-high-recall effect when adding all weakly labeled patterns, both leading to a sub-optimal F_1 measure. On the other hand, adding a limited amount of similar patterns may increase recall without harming precision too much. In this section, we investigate a selection strategy for the relations, how the quality of the training set depends on the fraction of similar patterns it is extended with. In our experimental setup, we start from the training set that only contains the $N_{filtered}$ instances that match the manually labeled patterns, gradually adding weakly labeled data, and each time training binary classifiers on the corresponding training set. We chose to let the additional data grow exponentially, which allows studying the effect of adding few extra instances initially, but extending towards the full weakly supervised training set of size N_{DS} in a limited number of cases. More specifically, in K experiments of adding additional instances, the intermediate training set size N_k at step k is given by

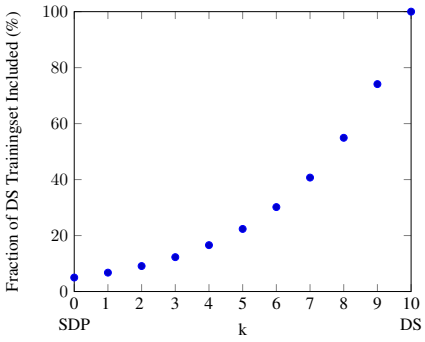


Figure 2.5: Example of the proposed sampling strategy for training set sizes, with $N_{filtered} = 0.05 N_{DS}$, and in $K = 10$ steps.

$$N_k = N_{filtered} \cdot \left(\frac{N_{DS}}{N_{filtered}} \right)^{k/K} \tag{2.4}$$

Figure 2.5 illustrates how an initial training set containing only 5% of the amount of instances from the full weakly labeled training set, is increased in $K = 10$ consecutive experiments.

Apart from studying the addition of varying amounts of similar patterns, in this section we also investigate the influence of the type of similarity measure used. In Section 2.3.2 we suggested the use of word embeddings, but is there a difference between different types of embeddings? Would embeddings work better than traditional dimension reduction techniques? And would such techniques indeed perform better than the original one-hot vector representations? These questions can be answered by considering several similarity measures. As a classical baseline, we represent SDPs using the average one-hot or bag-of-words (BOW) representations of the words contained in the SDPs. We also transform the set of one-hot representations using singular value decomposition (SVD) [65] fitted on the complete training set. For representations using the summed average of word embeddings described in Section 2.3.3, we use two sets of pre-trained *Word2Vec* embeddings[1] (trained on news text) and *GloVe* embeddings[2] (trained on Wikipedia text).

Figure 2.6 shows the effect of adding different amounts of weakly labeled data, for different values of k as in eq. 2.4 (with $K = 10$ steps) and for similarity measures based on the different types of representations described above. Six frequently occurring relations were selected such that they give an idea of the various forms of behavior that we observed dur-

[1] https://code.google.com/p/word2vec/
[2] http://nlp.stanford.edu/projects/glove/

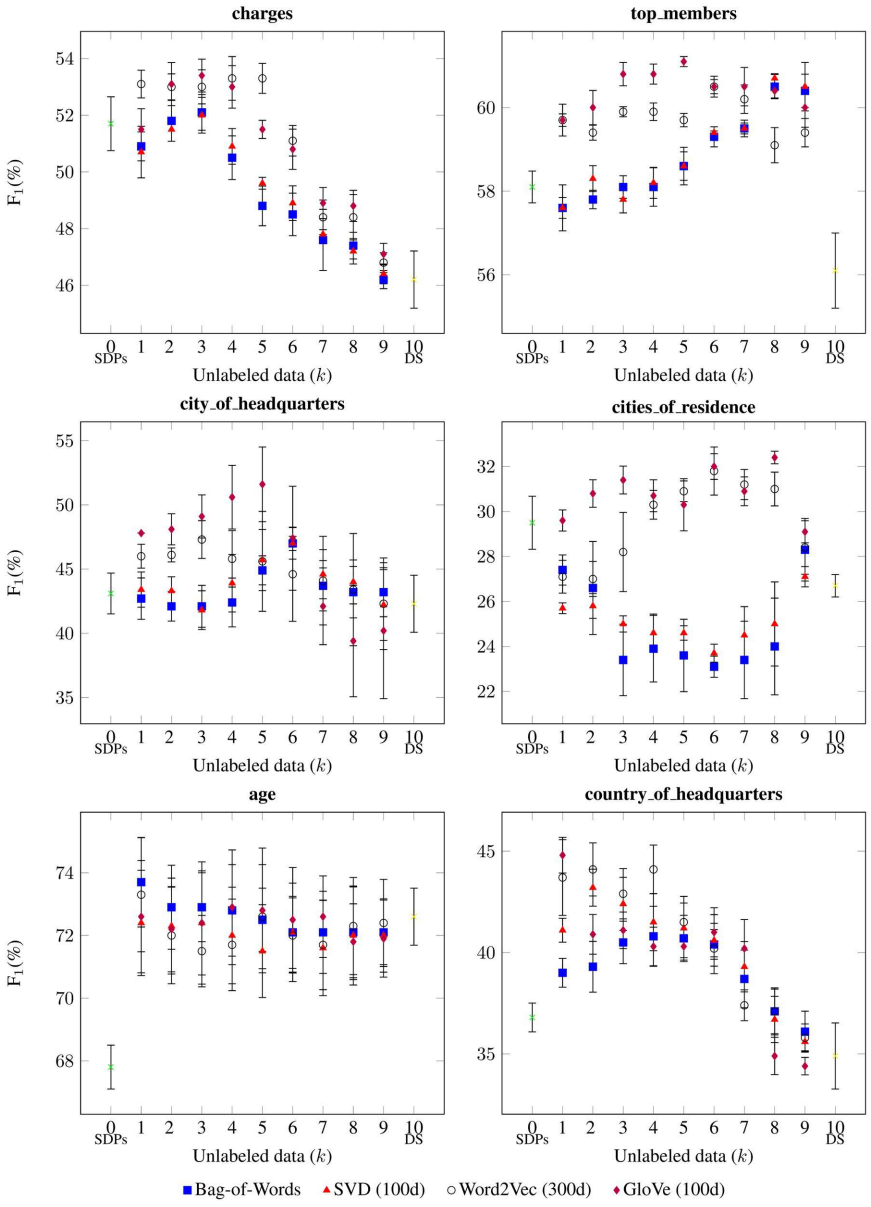


Figure 2.6: Illustration of the behavior of Semantic Label Propagation for different dimension reduction techniques, and different amounts of added weakly labeled data, quantified by k (as in eq. 2.4), with $K = 10$. $k = 0$ corresponds to only accepting manually filtered SDPs, and $k = 10$ corresponds to using all weakly labeled (DS) data for training.

ing our investigation of all extracted relations. The chosen effectiveness measure is the optimal F_1 value of classification on a development set, consisting of training data from 2012 and 2013. (In the next Section we will evaluate on a held-out test set, which consists of queries from the 2014 TAC ESF task, whereby the optimal value of k and type of dimension reduction is selected based on the development set.) Also shown are standard deviations on these optimal F_1-values, obtained by resampling different positive and negative instances for training the classifier. Several insights can be gained from Fig. 2.6:

- *SDPs vs full DS training set:* We observe that the effect of expanding the initial training set is strongly dependent on the specific relation and the quality of the initial training data. In many cases training data filtered using only highly confident SDPs ($k = 0$) generates a better relation extractor than pure DS ($k = K$). This holds for all shown relations, except for the *age* relation. We have to be aware that wrongly annotating an important pattern, or by chance missing any in the top most confident ones, can strongly reduce recall when only using the accepted SDPs. Adding even a small amount of similar patterns may hence result in a steep increase in effectiveness, such as for $k = 1$ in the *age* and *country_of_headquarters* relations.
- *Effect of semantic label propagation:* When relaxing the filtering (i.e., increasing k) by adding unlabeled data, the optimal F_1 tends to increase until a certain point, and then again drops towards the behavior of a fully DS training set, because the quality or similarity of the added training data declines and too many false positives are re-introduced. The threshold on the amount of added DS instances is thus an important parameter to tune on a development set. For some of the relations there is an optimal amount of added unlabeled data, whereas other relations show no clear optimum and fluctuate between distant and filtered classifiers' values.
- *Impact of dimensionality reduction:* The use of word embeddings often leads to an improved maximum F_1 value with respect to the BOW representations or SVD-based dimension reduction. This is for example very clear for the *charges*, *city_of_headquarters*, or *cities_of_residence* relations, with a slight preference of the *GloVe* embeddings with respect to *Word2Vec* for this application. However, we also noticed that word embeddings are not always better than the BOW or SVD based representations. For example, the highest optimal F_1 for the *age* relation is reached with the BOW model.

2.4.5 End-to-End Knowledge Base Population Results

This section presents the results of training binary relation classifiers according to our new strategy for each of the 41 relations of the TAC KBP

schema. We tuned hyperparameters on data of the 2012 and 2013 tracks and now test on the last edition of the ESF track of 2014.

Besides the thresholds for choosing the amount of unlabeled data added as discussed previously (i.e., the value of k), other parameters include regularization and the ratio between positive and negative instances, which appeared to be an important parameter influencing the confidence of an optimal F_1 value greatly. Different ratios of negative to positive instances resulted in shifting the optimal trade-off between precision and recall. The amount of available negative training data was on many occasions larger than the amount of available positive data. More negative than positive training data overall appeared to result in lower positive classification probabilities assigned by the classifier to test instances. Negative instances had to be down-weighted multiple times to prevent the classifier from being too strict and rarely classify a relation as true. For each relation, this parameter was tuned for optimal F_1 value at the 0.5 probability threshold of the logistic regression classifier.

We use the official TAC KBP evaluation script which calculates the micro-average of all classifications. All methods are evaluated while ignoring provenances (the character offsets in the documents which contain the justification for extraction of the relation), so as not to penalize any system for finding a new provenance not validated in the official evaluation key. A listing of precision, recall and F_1 for the top 20 most frequently occurring relations in the test set is shown in Table 2.4.

Besides traditional distant supervision (also known as *Mintz++* [29], indicated as 'distant Supervision' in Table 2.4), we compare our new semi-supervised approach ('Semantic Label Propagation') to a fully supervised classifier trained by manually labeling $50,000$ instances ('Fully Supervised'), and to the classifiers obtained by purely filtering on manually labeled patterns ('SDP Filtered'). We also use the fully supervised classifiers in a traditional self-training scheme, classifying distantly supervised instances in the complete feature space and adding confident instances to the training set ('Self-Training (Instances)'). The supervision needed for these classifiers required far more annotation effort than the feature certainty sampling of Semantic Label Propagation.

The official F_1 value of 36.4% attained using Semantic Label Propagation is equivalent to the second best entry out of eighteen submissions to the 2014 ESF track [22]. A relation extractor is but a part of a KBP system and is influenced by each of the other modules (e.g., recognition and disambiguation of named entities), which makes it hard to compare to other systems. This is the case for the absolute values of Table 2.4, but still, it demonstrates the overall quality of our relation extractors. Especially, our system relying on very limited annotations has a competitive place among systems that rely on many hours of manual feature engineering [49]. Comparing the results for Semantic Label Propagation with the other approaches shows that the proposed method that combines a small

labeling effort based on feature certainty with the Semantic Label Propagation technique, outperforms the DS method, semi-supervision using instance labeling, and full supervision methods. This is also confirmed in Fig. 2.7, which shows the trade-off between the precision and recall averaged over all TAC KBP relations for the different methods described above, using the TAC KBP evaluation script (varying the thresholds on classification).

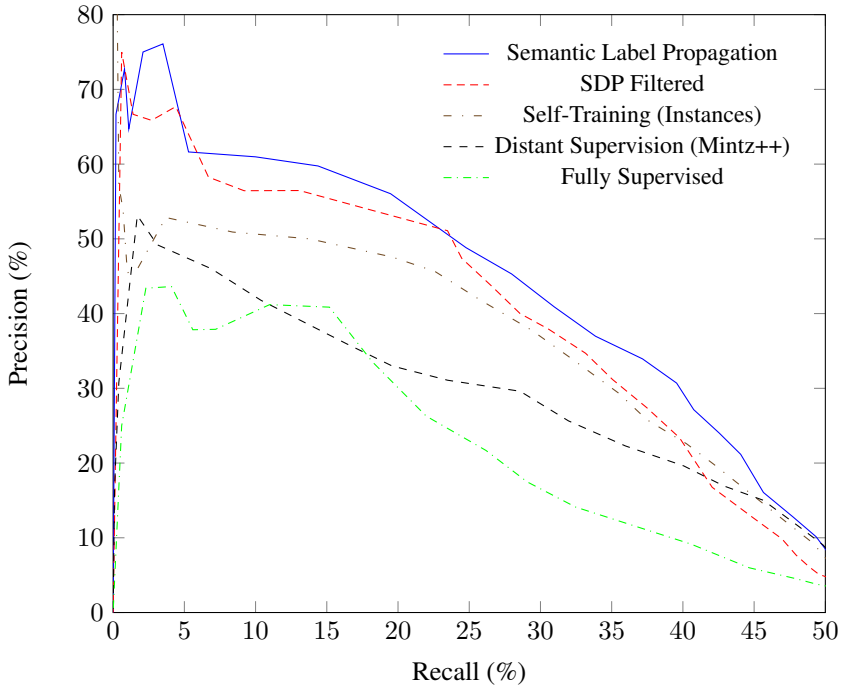


Figure 2.7: Precision-Recall Graph displaying the output of the TAC KBP evaluation script on different systems, for varying classifier decision thresholds.

One would expect the SDP filtered and fully supervised extractors to attain high precision, but this is not the case for some of the relations. For example, for relation *countries_of_residence* recall of these extractors is higher than recall of the SLP method. However, only those precision and recall scores are shown that correspond to the maximum values for $F_1$and while precision could have been higher for these extractors at the cost of lower recall, recall is equally important for this type of evaluation. The SDP filtered and fully supervised extractors are likely to attain high precision values, but this will not compensate for the loss in recall when evaluating F_1scores. We conclude by noting that the results may also be influenced to

Table 2.4: Results for Frequent Relations and official TAC-scorer

	Distant Supervision (Mintz++)			SDP Filtered			Fully Supervised			Self-Training (Instances)			Semantic Label Propagation		
Relation	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
title	22.3	58.8	32.3	36.1	39.1	37.5	28.0	61.1	38.4	36.5	43.2	**39.6**	37.3	41.2	39.2
top_members_employees	50.6	63.4	56.3	51.3	63.4	56.7	62.6	53.9	57.9	56.3	63.4	59.6	63.5	62.5	**63.0**
employee_or_member_of	31.4	34.0	32.6	33.8	51.0	**40.7**	23.5	45.7	31.0	32.2	40.4	35.8	27.9	51.0	36.1
age	71.6	72.5	72.0	75.6	70.0	72.7	68.0	62.5	64.9	73.6	70.0	71.8	68.8	82.5	**75.0**
origin	100.0	23.0	37.4	28.5	80.0	42.0	29.4	66.6	40.8	27.5	73.3	40.0	31.7	86.6	**46.4**
countries_of_residence	100.0	23.0	37.4	22.4	84.6	35.4	22.2	92.3	35.8	50.0	38.4	**43.4**	35.2	46.1	39.9
charges	45.0	52.9	48.6	40.9	52.9	46.1	70.4	44.1	**54.2**	47.6	58.8	52.6	44.3	68.1	53.7
cities_of_residence	22.9	45.8	30.5	31.5	25.0	27.9	11.2	62.5	19.0	36.3	16.6	22.8	34.4	41.6	**37.7**
cause_of_death	30.7	36.3	33.3	29.4	45.4	35.7	28.3	31.8	29.9	37.5	27.2	31.5	33.3	45.4	**38.4**
spouse	50.0	45.4	47.6	50.0	45.4	47.6	75.0	27.2	39.9	35.7	45.4	40.0	71.4	45.4	**55.5**
city_of_death	100.0	16.6	28.5	14.2	16.6	15.3	5.2	100.0	9.9	20.0	16.6	18.1	20.0	33.3	**25.0**
country_of_headquarters	22.7	41.6	29.4	62.5	41.6	**50.0**	25.0	50.0	33.3	100.0	25.0	40.0	100.0	33.3	**50.0**
date_of_death	66.6	50.0	**57.1**	66.6	50.0	**57.1**	50.0	25.0	33.3	66.6	50.0	**57.1**	66.6	50.0	**57.1**
(per:)parents	37.0	50.0	42.5	42.1	40.0	41.0	37.5	15.0	21.4	34.6	45.0	39.1	40.9	45.0	**42.9**
(org:)alternate_names	20.0	28.5	23.5	18.7	85.7	30.7	20.0	28.5	23.5	16.2	85.7	27.2	19.3	85.7	**31.5**
statesorprovinces_of_residence	50.0	55.5	**52.6**	50.0	44.4	47.0	53.5	44.4	48.5	45.4	55.5	49.9	50.0	44.4	47.0
founded_by	53.8	43.7	48.2	80.0	50.0	61.5	75.0	37.5	50.0	62.5	62.5	62.5	81.8	56.2	**66.6**
children	21.4	27.2	24.0	35.7	45.4	40.0	50.0	9.2	15.5	27.2	27.2	27.2	38.4	45.4	**41.6**
city_of_headquarters	42.8	100.0	59.9	46.1	66.6	54.5	36.3	88.8	51.5	46.6	77.7	58.3	71.4	55.5	**62.5**
siblings	100.0	28.5	**44.4**	100.0	28.5	**44.4**	100.0	14.2	24.9	66.6	28.5	39.9	100.0	28.5	**44.4**
(org:)parents	33.3	33.3	33.3	33.3	66.6	**44.4**	33.3	33.3	33.3	33.3	33.3	33.3	33.3	66.6	**44.4**
Official TAC Scorer (Micro-F_1)	29.3	28.1	28.7	35.5	33.7	34.7	22.7	26.0	24.3	37.5	29.4	33.0	36.9	35.9	**36.4**

peculiarities of the data. Entities chosen by TAC may not always be representative for the majority of persons or organizations in the training data: TAC entities are in many cases more difficult than the average entity from the training set and the most common way of expressing a relationship for these entities might not be present in the test set.

2.4.6 2015 TAC KBP Cold Start Slot Filling

The Slot filling task in TAC KBP in 2015 was organized as part of the Cold Start Slot Filling track, where the goal is to search the same document collection to fill in values for specific slots for specific entities, and in a second stage fill slots for answers given during the first stage. In the authors' TAC KBP 2015 submission [64], the ideas presented in this paper were applied, leading to a second place in the Slot Filling Variant. The results showed the influence of a clean training set and the effectiveness of self-training. A top-performing entry was again based on a database system similar to DeepDive [50] and training set filtering using high-precision patterns. We note that the idea of self-training using a first stage high-precision classifier was also included in this system, independently of the work presented in this paper. Some participants successfully used ensembles of neural architectures for relation extraction. However, a selection of our linear classifiers in combination with a careful filtering of distantly supervised training data was shown to outperform these more sophisticated ensembles.

2.5 Conclusions

In this paper we set out to create high quality training data for relation extractors for automatic knowledge base population systems, while requiring negligible amounts of supervision. To achieve this, we combine the following techniques for the unsupervised generation of training data and manual supervision: (i) *distant supervision (DS):* known relations from an existing knowledge base are used to automatically generate training data, (ii) *feature annotation:* rather than labeling instances, features (e.g., text patterns expressing a relationship) are annotated, selected by means of an active learning criterion based on confidence, and (iii) *semantic feature space representation:* low dimensional vector representations are used to detect additional, semantically related patterns that do not occur in the thus far selected training data, leaving useful patterns undetected otherwise. Thus, we address the problem of noisy training data obtained when using DS alone, by filtering of the training data using high-precision patterns to increase precision (see [52]). After this, to improve recall, we introduce the semi-supervised Semantic Label Propagation method, that allows relaxing the pattern-based filtering of the DS training data by again including weakly supervised items that are sufficiently "similar" to highly confident

instances. We found that a simple linear combination of the embeddings of words in a relation pattern is an effective representation when propagating labels from supervised to weakly supervised instances. Tuning a threshold parameter for similarity creates an improved training set for relation extraction.

The main contributions of this paper to the domain of relation extraction and automatic knowledge base population, are (i) the novel methodology of filtering an initial DS training set, where we motivated and demonstrated the effectiveness of an almost negligible manual annotation effort, and (ii) the Semantic Label Propagation model for again expanding the filtered set in order to increase diversity in the training data. We evaluated our classifiers on the knowledge base population task of TAC KBP and showed the competitiveness with respect to established methods that rely on a much heavier annotation cost.

2.A Using Active Learning and Semantic Clustering for Noise reduction in Distant Supervision

Previously we discussed a method for noise reduction using a single batch of supervision of labeled shortest dependency paths for the task of relation extraction. As an alternative direction for noise reduction, we apply cluster-aware active learning to distantly supervised training data. To improve the efficiency of the standard active learning procedure, we transform the text to a semantic vector space using a simple averaged embedding of the tokens in between the corresponding entities. We provide an evaluation for several frequently occuring relations.

⋆ ⋆ ⋆

L. Sterckx, T. Demeester, J. Deleu and C. Develder

Presented at the Fourth Workshop on Automated Base Construction at NIPS2014, Proceedings. p.1-6, Montreal, Canada, 2014.

Abstract The use of external databases to generate training data, also known as Distant Supervision, has become an effective way to train supervised relation extractors but this approach inherently suffers from noise. In this paper we propose a method for noise reduction in distantly supervised training data, using a discriminative classifier and semantic similarity between the contexts of the training examples. We describe an active learning strategy which exploits hierarchical clustering of the candidate training samples. To further improve the effectiveness of this approach, we study the use of several methods for dimensionality reduction of the training samples. We find that semantic clustering of training data combined

with cluster-based active learning allows filtering the training data, hence facilitating the creation of a clean training set for relation extraction, at a reduced manual labeling cost.

2.A.1 Introduction

For the task of extracting relations between entities according to a fixed schema (also known as Knowledge Base Population (KBP)), distantly supervised approaches are currently state-of-the-art [66]. A requisite for the effectiveness of these techniques is the availability of labeled data, which is expensive to obtain. An approach to solve this issue and produce large quantities of training data is distant supervision (DS) [29]. DS creates labeled data using readily available repositories like FreeBase or DBpedia with facts like "*Person* $\rightarrow$ *city-of-residence*$\rightarrow$ *Location*" (for the remainder denoted as "*per:city_of_residence*") and the assumption that every phrase mentioning both entities participating in the relation expresses the corresponding relation from the database. Using this approach, a large quantity of training data can be generated automatically. However, intuitively this assumption only holds for a fraction of the extracted mentions, as two entities may co-occur in one sentence for many alternative reasons. A challenge we address in this paper is to develop strategies to improve the quality of the training data and reduce the amount of noise.

In our participation in the Text Analysis Conference for Knowledge Base Population (TAC-KBP) slot filling track organized by NIST [67], a baseline supervised classification using DS was implemented as described in [68]. In our submissions, we already showed the value of noise reduction based on straightforward human annotation of randomly selected training instances: cleaning based on a classifier (trained on the annotated instances) resulted in 8% higher precision. As this required extra manual annotation of the training samples, we search for an efficient way to query the distantly supervised data and train a classifier using a minimal amount of supervision but an improved noise reduction.

This work contributes by presenting a strategy for noise reduction using a supervised classifier trained using labeled mentions from distantly supervised data. By incorporating semantic relatedness between the mentions we can use an active learning approach which exploits the resulting clustering of training data. Intelligent querying of training data clusters and assigning labels to similar unknown training examples trains a classifier based on less human supervision while optimizing the capability of separating noisy from true relation contexts.

2.A.2 Related Work

The approach of DS was first presented by Mintz et al. [29] for training of binary Support Vector Machines which used a set of lexical and non-lexical

features for classification. Since then, several methods for noise reduction of the data have been proposed. For a recent survey we refer to Roth et al. [69]. Models like the one proposed by Riedel et al. [24], MultiR [25] and MIML-RE [70] involve latent variables which model the assumption that at least one generated example for an entity pair and relation is a true positive, or apply a generative model [45]. Our approach is less complex, using a discriminative classifier based on manually annotated examples of true positive and false positive relation mentions within each of the generated training sets. Using this classifier, we filter training data explicitly, independent of the entities involved and for each relation separately, solely based on the surface text.

Recent work has combined DS with small amounts of labeled data, these labels are either included directly in a latent variable model [71] or used in an active learning setting. Active learning was previously performed in relation extraction by Sun and Grishman [72] for extending a relation extraction system to recognize a new type of relation. An approach which uses active learning for DS was recently proposed by Angeli et al. [73] and successfully applied in the top performing system in the TAC slot filling competition [66]: they show that a small number of examples can yield improvements in end-to-end accuracy of the relation extraction using several approaches from active learning literature and a new metric incorporating uncertainty and representativeness. Our work differs from this and others in that we use a cluster based active learning approach, evaluating directly on a set of labeled training examples.

2.A.3 Semantic-Cluster-Aware Sampling

Our approach assumes that true positive mentions within each training set are similar to each other, in terms of text and meaning, and tend to cluster together, unlike false positive mentions which are less similar and more diverse. This inspired us for the application of cluster-aware sampling of the data for training of the noise-reduction classifier. An active learning approach that exploits cluster structure in data was introduced by Dasgupta and Hsu [74]. This algorithm takes a pool of unlabeled data and performs hierarchical clustering, resulting in a tree structure of data points. The algorithm starts out by randomly querying data points in the vector space and searches for a pruning of the tree that optimizes the pureness of each cluster. Each iteration, a number of data points are sampled in such a way that less pure clusters are more likely to be sampled from and unseen samples receive the label of the majority of known samples in the cluster they belong to.

As stated in the original paper [74], the algorithm is most effective when pure clusters are found at the top of the hierarchical tree. Thus, when applied to noise reduction, this approach benefits from relation contexts that are clustered according to the meaning or relation they express. The simple

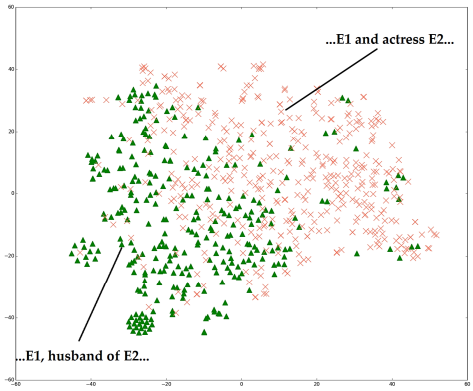


Figure 2.8: Visualization of relation contexts in a semantic vector space for relation *"per:spouse_of"*.

bag-of-words representation results in a high dimensionality of the relation contexts with only few ways of clustering contexts with similar meaning. We need a transformation of the contexts into a vector space of reduced dimension, with those having a similar expression of relation being transformed into similar representations. This is exactly what semantic clustering achieves, i.e. clustering contexts according to meaning.

Semantic clustering of relations has been performed on several occasions in the context of Open Information Extraction to cluster output having similar meaning and is related to the task of paraphrase and synonym detection [75–78]. We use a transformation based on a simple composition of the words participating in the context. While much research has been directed at ways of constructing distributional representations of individual words, for example co-occurrence based representations and word embeddings, there has been far less consensus regarding the representation of larger constructions such as phrases and sentences from these representations. Blacoe et al. [59] show that a simple composition like addition or multiplication of the distributional word representations is competitive with more complex operations like the Recursive Neural Networks proposed by Socher et al. [79] for detection of paraphrases and synonyms. We chose to ignore word order and sum all distributional representations from words participating in the surrounding context of the mention and normalize them.

2.A.4 Experiments and Results

We use FreeBase [80] as our source of fact relations by first matching the schema from the TAC-KBP to fields from FreeBase. The participating enti-

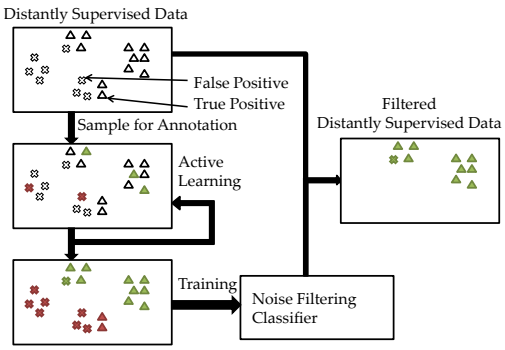


Figure 2.9: Methodology for filtering of noisy mentions.

ties are matched in phrases from articles from the English GigaWord corpus [60]. As part of a participation in the TAC-KBP slot filing competition, a team of students was asked to assign 2.000 training samples with a *True* or *False* label with respect to the 2014 TAC-annotation guidelines for a selection of 12 relations with a large quantity of training data. As these samples were selected at random, some of the relations contained very pure or highly noisy training sets. Phrases were filtered for duplicates and entity names were removed from the surface text.

Effective representations should be able to separate true examples of a relation being expressed from false examples. We visualize this in Figure 1 for the relation *per:spouse_of*. Words in between the subject and object entities of the relation are transformed to their semantic vectors using word embeddings which are summed and normalized. In our experiments we use the GloVe word-embeddings with 100 dimensions trained on Wikipedia text [81], which are made available from the authors' website.[1] The resulting sentence representations are clustered and represented in a two-dimensional space using the t-SNE algorithm [82] in Figure 1. True examples of the relation are represented in Figure 1 as dark triangles, while false examples are the lighter crosses. The resulting figure shows that this basic transformation alone is able to capture some of the semantic meaning of relations.

The active learning strategy is performed on 70% of the DS-data, 30% is set aside to evaluate classification. The general methodology for filtering distantly supervised data is shown in Figure 2. Previously described active learning iteratively only queries a number of DS-examples, but results in a fully labeled distantly supervised data set (each unknown sample then receives the label of the majority of its cluster). The resulting fully labeled DS-data is used to train a logistic regression classifier using only

[1] http://nlp.stanford.edu/projects/glove/

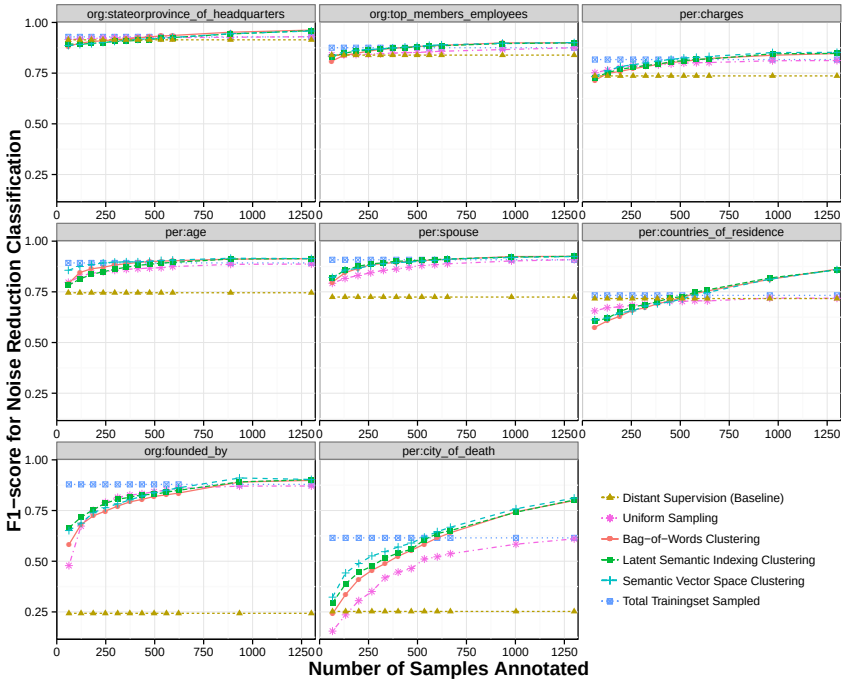


Figure 2.10: Performance of cluster-based active learning approach.

word count vectors as features in a basic text classification setting to filter the training data. At most two words before the entity first mentioned, the words in between the entities and at most two words following the entity mentioned last are included. We compare our cluster-based active learning in the semantic vector space, with uniform sampling, clustering using Bag-of-Words vectors and clustering after transformation using Latent Semantic Indexing (LSI) [83] (also for 100 dimensions). This process is repeated 20 times using stratified cross-folds.

Using all of the labeled data, supervised noise reduction is able to increase average fraction of true positive of the DS training set from 47% to 84% while maintaining a recall of 88%. Noise reduction using active learning for a selection of relations is presented in Figure 3. Separately for each relation, after each increase of 5% sampled data we calculate the averaged F1 score of the classification for each of the strategies. Performance of the noise reduction is highly dependent on the relation. For relatively pure training sets, as is the case for relation "*org:state_or_province_of_headquarters*", with more than 85% of the training data being positive examples, are hard to filter. For these relations supervised filtering appears ineffec-

Table 2.5: Macro-average filter performance using 70 labeled distantly supervised training examples

	Precision	**Recall**	$\mathbf{F_1}$
Distant Supervision (Baseline)	51.9	100.0	60.8
Random Sampling	72.0	**72.8**	66.0
Bag-of-Words Clustering	73.4	65.2	66.6
Latent Semantic Indexing Clustering	73.7	68.5	68.3
Semantic Vector Space Clustering	**74.6**	71.4	**71.2**

tive or even detrimental, others need a minimum amount of samples to benefit like "*per:countries_of_residence*". Cluster aware active learning is an effective strategy for a majority of the noisy relations, converging faster to the optimal performance of filtering. Overall, performance using semantic clustering of contexts is slightly better than using LSI clustering, while with very few samples and relations like "*per:age*" and "*per:city_of_death*" performance increase is larger. Another observation is that, because the algorithm also provides the test samples with a label (based on the majority of the labels in the same cluster as the test sample), classification performance surpasses that of a fully labeled training set while approximately only half is sampled. Table 1 shows macro average precision, recall and F1 using a minimal amount of only 70 samples for noisy relations (fraction of true positives less than 85%).

2.A.5 Conclusion

In this paper we presented a novel approach for filtering a distantly supervised training set by building a binary classifier to detect true relation mentions, the classifier is trained using a cluster based active learning strategy. We show that clustering of relation mentions and adding semantic information reduces human effort and makes this a promising approach more feasible to filter a wide variety of relations. For future work we suggest the use of more sophisticated methods which take into account composition for transforming context to a semantic vector space.

References

[1] F. M. Suchanek, G. Kasneci, and G. Weikum. *Yago: a core of semantic knowledge*. In Proceedings of the 16th international conference on World Wide Web, pages 697–706. ACM, 2007.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. *DBpedia-A crystallization point for the Web of Data*. Web Semantics: science, services and agents on the world wide web, 7(3):154–165, 2009.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. *Freebase: a collaboratively created graph database for structuring human knowledge*. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250. ACM, 2008.

[4] T. M. Mitchell, W. W. Cohen, E. R. H. Jr., P. P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. T. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. *Never-Ending Learning*. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., pages 2302–2310, 2015.

[5] N. Nakashole, M. Theobald, and G. Weikum. *Scalable knowledge harvesting with high precision and high recall*. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 227–236. ACM, 2011.

[6] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. *Textrunner: open information extraction on the web*. In Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pages 25–26. Association for Computational Linguistics, 2007.

[7] J. Fan, D. Ferrucci, D. Gondek, and A. Kalyanpur. *Prismatic: Inducing knowledge from a large scale lexicalized relation resource*. In Proceedings of the NAACL HLT 2010 first international workshop on formalisms and methodology for learning by reading, pages 122–127. Association for Computational Linguistics, 2010.

[8] R. Speer and C. Havasi. *Representing General Relational Knowledge in ConceptNet 5*. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012, pages 3679–3686, 2012.

[9] E. Cambria, D. Olsher, and D. Rajagopal. *SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis*. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada., pages 1515–1521, 2014.

[10] S. Poria, E. Cambria, A. F. Gelbukh, F. Bisio, and A. Hussain. *Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns*. IEEE Comp. Int. Mag., 10(4):26–36, 2015.

[11] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. *Knowledge vault: A web-scale approach to probabilistic knowledge fusion*. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 601–610. ACM, 2014.

[12] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. *Building, Maintaining, and Using Knowledge Bases: A Report from the Trenches*. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, pages 1209–1220, New York, NY, USA, 2013. ACM.

[13] B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek. *Distant Supervision for Relation Extraction with an Incomplete Knowledge Base*. In Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA, pages 777–782, 2013.

[14] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. *A novel use of statistical parsing to extract information from text*. In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pages 226–233. Association for Computational Linguistics, 2000.

[15] N. Kambhatla. *Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations*. In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, page 22. Association for Computational Linguistics, 2004.

[16] E. Boschee, R. Weischedel, and A. Zamanian. *Automatic information extraction*. In Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA, pages 2–4. Citeseer, 2005.

[17] J. Jiang and C. Zhai. *A Systematic Exploration of the Feature Space for Relation Extraction*. In Human Language Technology Conference of the

North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA, pages 113–120, 2007.

[18] A. Sun, R. Grishman, and S. Sekine. *Semi-supervised relation extraction with large-scale word clustering*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 521–529. Association for Computational Linguistics, 2011.

[19] R. C. Bunescu and R. J. Mooney. *A shortest path dependency kernel for relation extraction*. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 724–731. Association for Computational Linguistics, 2005.

[20] E. Agichtein and L. Gravano. *Snowball: Extracting relations from large plain-text collections*. In Proceedings of the fifth ACM conference on Digital libraries, pages 85–94. ACM, 2000.

[21] S. Gupta and C. D. Manning. *SPIED: Stanford Pattern-based Information Extraction and Diagnostics*. Proceedings of the ACL 2014 Workshop on Interactive Language Learning, Visualization, and Interfaces (ACL-ILLVI), 2014.

[22] M. Surdeanu and H. Ji. *Overview of the english slot filling track at the tac2014 knowledge base population evaluation*. Proc. Text Analysis Conference (TAC2014), 2014.

[23] J. Shin, S. Wu, F. Wang, C. De Sa, C. Zhang, and C. Ré. *Incremental Knowledge Base Construction Using DeepDive*. Proceedings of the VLDB Endowment, 8(11):1310–1321, 2015.

[24] S. Riedel, L. Yao, and A. McCallum. *Modeling relations and their mentions without labeled text*. In Machine Learning and Knowledge Discovery in Databases, pages 148–163. Springer Berlin Heidelberg, 2010.

[25] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. *Knowledge-based weak supervision for information extraction of overlapping relations*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 541–550. Association for Computational Linguistics, 2011.

[26] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. *Multi-instance multi-label learning for relation extraction*. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 455–465. Association for Computational Linguistics, 2012.

[27] G. Angeli, J. Tibshirani, J. Wu, and C. D. Manning. *Combining Distant and Partial Supervision for Relation Extraction*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1556–1567, 2014.

[28] M. Pershina, B. Min, W. Xu, and R. Grishman. *Infusion of Labeled Data into Distant Supervision for Relation Extraction*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 732–738, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[29] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. *Distant supervision for relation extraction without labeled data*. In ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, pages 1003–1011, 2009.

[30] D. Zelenko, C. Aone, and A. Richardella. *Kernel Methods for Relation Extraction*. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, pages 71–78, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[31] A. Culotta and J. Sorensen. *Dependency tree kernels for relation extraction*. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, page 423. Association for Computational Linguistics, 2004.

[32] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. *Relation Classification via Convolutional Deep Neural Network*. In COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland, pages 2335–2344, 2014.

[33] K. Xu, Y. Feng, S. Huang, and D. Zhao. *Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 536–540, 2015.

[34] H. Adel, B. Roth, and H. Schütze. *Comparing Convolutional Neural Networks to Traditional Models for Slot Filling*. CoRR, abs/1603.05157, 2016.

[35] S. Brin. *Extracting Patterns and Relations from the World Wide Web.* Technical Report 1999-65, Stanford InfoLab, November 1999.

[36] C. Zhang, W. Xu, Z. Ma, S. Gao, Q. Li, and J. Guo. *Construction of semantic bootstrapping models for relation extraction*. Knowledge-Based Systems, 83:128–137, 2015.

[37] M. Komachi, T. Kudo, M. Shimbo, and Y. Matsumoto. *Graph-based Analysis of Semantic Drift in Espresso-like Bootstrapping Algorithms*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 1011–1020, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[38] J. R. Curran, T. Murphy, and B. Scholz. *Minimising semantic drift with mutual exclusion bootstrapping*. Proceedings of the Conference of the Pacific Association for Computational Linguistics, pages 172–180, 2007.

[39] D. S. Batista, B. Martins, and M. J. Silva. *Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 499–504, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[40] M. Craven and J. Kumlien. *Constructing Biological Knowledge Bases by Extracting Information from Text Sources*. In Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, August 6-10, 1999, Heidelberg, Germany, pages 77–86, 1999.

[41] I. Augenstein, A. Vlachos, and D. Maynard. *Extracting Relations between Non-Standard Entities using Distant Supervision and Imitation Learning*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 747–757, 2015.

[42] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. *Knowledge-based Weak Supervision for Information Extraction of Overlapping Relations*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 541–550, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[43] A. Intxaurrondo, M. Surdeanu, O. L. de Lacalle, and E. Agirre. *Removing noisy mentions for distant supervision*. Procesamiento del lenguaje natural, 51:41–48, 2013.

[44] E. Alfonseca, K. Filippova, J.-Y. Delort, and G. Garrido. *Pattern learning for relation extraction with a hierarchical topic model*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 54–59. Association for Computational Linguistics, 2012.

[45] S. Takamatsu, I. Sato, and H. Nakagawa. *Reducing wrong labels in distant supervision for relation extraction*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pages 721–729. Association for Computational Linguistics, 2012.

[46] J. Chen, D. Ji, C. L. Tan, and Z. Niu. *Relation extraction using label propagation based semi-supervised learning*. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 129–136. Association for Computational Linguistics, 2006.

[47] C. Zhang, F. Niu, C. Ré, and J. Shavlik. *Big data versus the crowd: Looking for relationships in all the right places*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pages 825–834. Association for Computational Linguistics, 2012.

[48] H. Ji and R. Grishman. *Knowledge base population: Successful approaches and challenges*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 1148–1158. Association for Computational Linguistics, 2011.

[49] G. Angeli, S. Gupta, M. Jose, C. D. Manning, C. Ré, J. Tibshirani, J. Y. Wu, S. Wu, and C. Zhang. *StanfordâĂŹs 2014 slot filling systems*. TAC KBP, 2014.

[50] C. Zhang. *DeepDive: A Data Management System for Automatic Knowledge Base Construction*. PhD thesis, UW-Madison, 2015.

[51] H. S. Seung, M. Opper, and H. Sompolinsky. *Query by committee*. In Proceedings of the fifth annual workshop on Computational learning theory, pages 287–294. ACM, 1992.

[52] L. Sterckx, T. Demeester, J. Deleu, and C. Develder. *Using Active Learning and Semantic Clustering for Noise Reduction in Distant Supervision*. In 4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014), pages 1–6, 2014.

[53] G. Druck, B. Settles, and A. McCallum. *Active learning by labeling features*. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, pages 81–90. Association for Computational Linguistics, 2009.

[54] J. Attenberg, P. Melville, and F. Provost. *A unified approach to active dual supervision for labeling features and examples*. In In European conference on Machine learning and knowledge discovery in databases, pages 40–55, 2010.

[55] Z. Harris. *Distributional structure*. Word, 10(23):146–162, 1954.

[56] J. H. Martin and D. Jurafsky. *Speech and language processing*. International Edition, 2015.

[57] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. CoRR, abs/1301.3781, 2013.

[58] J. Pennington, R. Socher, and C. D. Manning. *Glove: Global Vectors for Word Representation*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1532–1543, 2014.

[59] W. Blacoe and M. Lapata. *A Comparison of Vector-based Representations for Semantic Composition*. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 546–556, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[60] D. Graff, J. Kong, K. Chen, and K. Maeda. *English gigaword*. Linguistic Data Consortium, 2003.

[61] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. *The Stanford CoreNLP natural language processing toolkit*. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60, 2014.

[62] M. Feys, L. Sterckx, L. Mertens, J. Deleu, T. Demeester, and C. Develder. *Ghent University-IBCN participation in TAC-KBP 2014 slot filling and cold start tasks*. In 7th Text Analysis Conference, Proceedings, pages 1–10, 2014.

[63] M. Stevenson and M. A. Greenwood. *Comparing information extraction pattern models*. In Proceedings of the Workshop on Information Extraction Beyond The Document, pages 12–19. Association for Computational Linguistics, 2006.

[64] L. Sterckx, J. Deleu, T. Demeester, and C. Develder. *Ghent University-IBCN participation in TAC-KBP 2015 cold start task*. In 8th Text Analysis Conference, Proceedings (To Appear), 2015.

[65] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. *Indexing by Latent Semantic Analysis*. JASIS, 41(6):391–407, 1990.

[66] M. Surdeanu and H. Ji. *Overview of the English Slot Filling Track at the TAC2014 Knowledge Base Population Evaluation*. 2014.

[67] *Task Description for English Slot Filling at TAC-KBP*. 2014.

[68] M. Feys, L. Sterckx, L. Mertens, J. Deleu, T. Demeester, and C. Develder. *Ghent University-IBCN Participation in TAC-KBP 2014 Slot Filling and Cold Start Tasks*. 2014.

[69] B. Roth, T. Barth, M. Wiegand, and D. Klakow. *A Survey of Noise Reduction Methods for Distant Supervision*. In Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, AKBC '13, pages 73–78, New York, NY, USA, 2013. ACM. Available from: http://doi.acm.org/10.1145/2509558.2509571, doi:10.1145/2509558.2509571.

[70] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. *Multi-instance multi-label learning for relation extraction*. pages 455–465, 2012.

[71] M. Pershina, B. Min, W. Xu, and R. Grishman. *Infusion of Labeled Data into Distant Supervision for Relation Extraction*. In Proceedings of the 2014 Conference of the Association for Computational Linguistics (ACL 2014), Baltimore, US, June 2014. Association for Computational Linguistics.

[72] A. Sun and R. Grishman. *Active Learning for Relation Type Extension with Local and Global Data Views*. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pages 1105–1112, New York, NY, USA, 2012. ACM. Available from: http://doi.acm.org/10.1145/2396761.2398409, doi:10.1145/2396761.2398409.

[73] G. Angeli, J. Tibshirani, J. Wu, and C. Manning. *Combining Distant and Partial Supervision for Relation Extraction*. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014. Available from: http://nlp.stanford.edu/pubs/2014emnlp-kbpactivelearning.pdf.

[74] S. Dasgupta and D. Hsu. *Hierarchical sampling for active learning*. In Proceedings of the 25th international conference on Machine learning, 2008.

[75] T. Hasegawa, S. Sekine, and R. Grishman. *Discovering relations among named entities from large corpora*. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004.

[76] W. Wang, R. BesanÃğon, O. Ferret, and B. Grau. *Semantic Clustering of Relations between Named Entities*. In A. PrzepiÃşrkowski and M. Ogrodniczuk, editors, Advances in Natural Language Processing, volume 8686 of *Lecture Notes in Computer Science*, pages 358–370. Springer International Publishing, 2014.

[77] L. Yao, A. Haghighi, S. Riedel, and A. McCallum. *Structured relation discovery using generative models*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011.

[78] L. Romano, M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli. *Investigating a generic paraphrase-based approach for relation extraction*. 2006.

[79] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. *Semantic Compositionality Through Recursive Matrix-vector Spaces*. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012.

[80] Google. *Freebase Data Dumps*. https://developers.google.com/freebase/data, 2014.

[81] J. Pennington, R. Socher, and C. D. Manning. *GloVe: Global Vectors for Word Representation*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014.

[82] L. Van der Maaten and G. Hinton. *Visualizing data using t-SNE*. Journal of Machine Learning Research, 2008.

[83] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. *Indexing by latent semantic analysis*. volume 41, pages 391–407, 1990.

3

Weakly Supervised Evaluation of Topic Models

In this chapter we present a new method for the evaluation of unsupervised topic models at reduced human supervision. Topic models such as Latent Dirichlet Allocation lead to reasonable statistical models of documents, but they offers no guarantee of producing results that are interpretable by humans and require a thorough evaluation of the output. Existing methods evaluate statistical goodness-of-fit, which offer no guarantee of interpretablity which would in turn require a costly annotation effort. We use small labeled collections as reference topics and present a measure for topic quality which correlates well with human evaluation. This way low-quality topics can be filtered out and the time needed for manual inspection of topics is reduced considerably.

⋆⋆⋆

L. Sterckx, T. Demeester, J. Deleu, L. Mertens and C. Develder

Presented at the European Conference on Information Retrieval, Amsterdam, Netherlands, 2014.

Abstract This paper presents an evaluation method for unsupervised topic models. How useful are topic models based on song lyrics for applications in music information retrieval? Unsupervised topic models on text corpora are often difficult to interpret. Based on a large collection of lyrics, we

investigate how well automatically generated topics are related to manual topic annotations.

We propose to use the kurtosis metric to align unsupervised topics with a reference model of supervised topics. This metric is well-suited for topic assessments, as it turns out to be more strongly correlated with manual topic quality scores than existing measures for semantic coherence. We also show how it can be used for a detailed graphical topic quality assessment.

3.1 Introduction

This paper presents an analysis of how well topic models can be used to detect lyrical themes for use in Music Information Retrieval (MIR), an interdisciplinary science developing techniques including music recommendation.

Probabilistic topic models are a tool for the unsupervised analysis of text, providing both a predictive model of future text and a latent topic representation of the corpus. Latent Dirichlet Allocation (LDA) is a Bayesian graphical model for text document collections represented by bags-of-words [1]. In a topic model, each document in the collection of documents is modeled as a multinomial distribution over a chosen number of topics, each topic is a multinomial distribution over all words. We evaluate the quality and usefulness of topic models for new music recommendation applications.

Although lyricism and themes are undeniably contributing to a musical identity, they are often treated as mere secondary features, e.g., for obtaining music or artist similarity, which are dominantly determined by the audio signal. Nevertheless, previous works have analyzed lyrics, mainly aimed at determining the major themes they address. Mahadero et al. [2] performed a small scale evaluation of a probabilistic classifier, classifying lyrics into five manually applied thematic categories. Kleedorfer et al. [3] focused on topic detection in lyrics using an unsupervised statistical model called Non-negative Matrix Factorization (NMF) on 32,323 lyrics.

After clustering by NMF, a limited evaluation was performed by a judgment of the most significant terms for each cluster. We expand on this work by performing a large-scale evaluation of unsupervised topic models using a smaller dataset of labeled lyrics and a supervised topic model.

While state-of-the-art unsupervised topic models lead to reasonable statistical models of documents, they offer no guarantee of producing results that are interpretable by humans and require a thorough evaluation of the output. When considering lyrics, there is no general consensus on the amount and nature of the main themes, as opposed to news-corpora (sports, science,...). A useful topic model for MIR, appends the music with a representation of the thematic composition of the lyrics. For use in applications like music recommendation, playlist generation, ..., the topics

should be interpretable. Evaluation methodologies based on statistical [1] or coherence [4] measures are not optimal for this purpose since they do not account for interpretability and relevance to the application. Chuang et al. [5] introduced a framework for the large-scale assessment of topical relevance using supervised topics and alignment between unsupervised and supervised topics.

Our contributions presented in this paper apply and build on aforementioned work, by assessing quality of unsupervised topics for use in MIR, and by introducing a new method for measuring and visualizing the quality of topical alignment, based on the kurtosis of the similarity between unsupervised topics and a reference set of supervised topics.

In Section 3.2, we present the data and our experimental set-up. The main topic model analysis is presented in Section 3.3, followed in Section 3.4 by conclusions.

3.2 Experimental Setup

The main dataset used for this research is the 'Million Song Dataset' (MSD) [6], with metadata for one million songs, and lyrics as bags-of-words for a subset of 237,662 songs from a commercial lyrics catalogue, 'musiXmatch'.

LDA was applied on the set of lyrics, using the Java-based package MALLET [7]. Three topic models were inferred from the subset of 181,892 English lyrics for evaluation, one with 60 (T60), 120 (T120) and 200 (T200) topics. A manual quality assessment of all of these topics was performed, with scores ranging from 1 (useless) to 3 (highly useful).

As an additional resource, a clean dataset of labels was provided by the website, 'GreenbookofSongs.com®'[1] (GOS), a searchable database of songs categorized by subject. This dataset contains 9,261 manually annotated song lyrics matched with the MSD (a small subsample of the GOS' complete database), with multiple labels from a large class-hierarchy of 24 super-categories with a total of 877 subcategories. Labeled Latent Dirichlet Allocation (L-LDA) is a variation of LDA for labeled corpora by incorporating user supervision in the form of a one-to-one mapping between topics and labels [8]. An L-LDA model with 38 supervised topics was inferred from the much smaller set of GOS-labeled lyrics, based on the GOS super-categories (but with the omission of minor categories like 'Tools', and splitting up of major categories like 'Love'). These are high-quality topics, but because of the limited size of the GOS data set, less representative for the entire scope of themes in the complete MSD lyrics collection.

[1] http://www.greenbookofsongs.com, the authors would like to thank Lauren Virshup and Jeff Green for providing access to the GOS-database

3.3 Topic Model Assessment

We define the suitability of topic models for use in MIR as determined by the amount of relevant and interpretable topics they produce for MIR. We first introduce suitable measures to evaluate to what extent unsupervised topics can be mapped to supervised topics obtained from tagged documents. We then show how these can be used as a better measure for the interpretability of topics than an existing alternative, and provide a visual analysis of topical alignment.

3.3.1 Measuring Topical Alignment

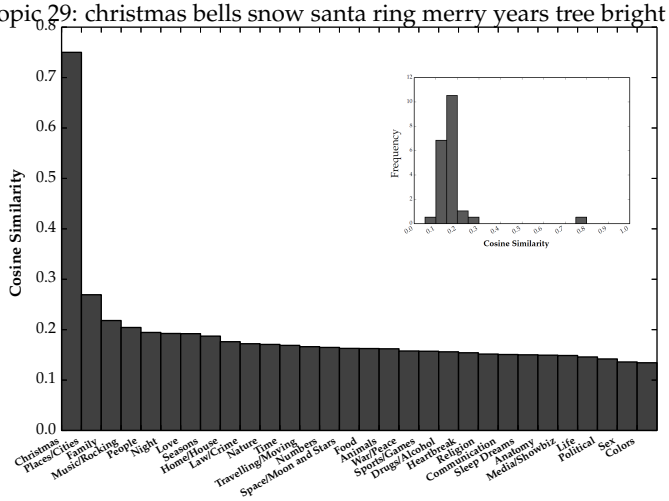
We define high-quality topics as topics for which a human judge finds a clear semantic coherence between the relevant terms in relation to an underlying concept (such as 'Love', or 'Christmas'). Such concepts are made explicit by an L-LDA model based on tagged documents, and we detect high-quality LDA-topics as those that bear a strong resemblance with L-LDA topics. For an unsupervised topic to represent a highly distinctive theme, ideally it should be highly similar to only a single supervised topic. For each of the unsupervised LDA-topics, the cosine similarity between the word-topic probability distribution is calculated with the distribution of each L-LDA topic.

We introduce two measures to assess the distribution of these similarities per LDA-topic, which measure how strongly the variance of the mean cosine similarity depends on extreme values (in this case, because of similarities that are much higher than the average). The first is the excess kurtosis (γ_2), traditionally used to detect peakedness and heavy tails [9]. The second is the normalized maximum similarity ($z_{\max}$), used in several outlier detection strategies.

$$\gamma_2 = \frac{\mu_4}{\sigma^4} - 3, \qquad z_{\max} = \frac{X_{\max} - \mu}{\sigma} \tag{3.1}$$

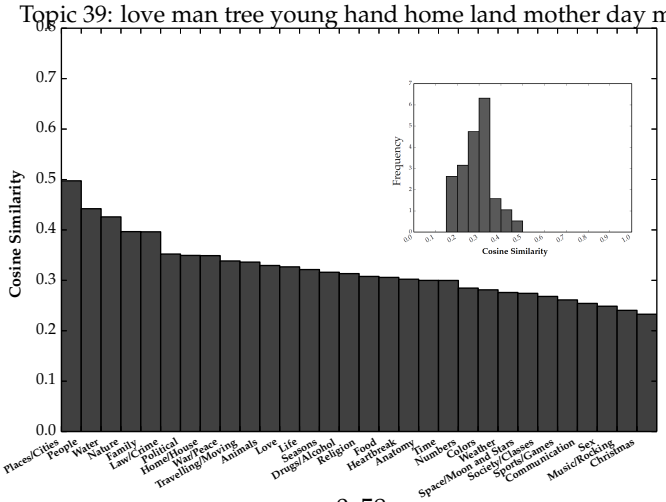
with μ_4 the fourth moment about the mean μ, σ the standard deviation, and $X_{\max}$ the maximum similarity. Figure 3.1 shows the similarities with the unsupervised topics for the high-quality LDA-topic 29, with a clearly matched supervised topic (and high values for γ_2 and $z_{\max}$), and for the low-quality LDA-topic 39 (with low γ_2 and $z_{\max}$). The insets show the histograms of the similarities. Various other metrics were evaluated as well, but with a lower ability of detecting the interesting distributions.

Topic 29: christmas bells snow santa ring merry years tree bright sleigh



$$\gamma_2 = 17,29$$
$$z_{\max} = 5.75$$

Topic 39: love man tree young hand home land mother day men



$$\gamma_2 = 0,58$$
$$z_{\max} = 2.70$$

Figure 3.1: Kurtosis measure and Normalized Maximum Similarity for topic evaluation

3.3.2 Semantic Coherence

A second evaluation was performed using metrics presented in [4], where the authors show that measures for semantic coherence are highly correlated with human quality judgments. These metrics use WordNet, a lexical

Table 3.1: Spearman correlations with manual quality-scores for the three topic models

Evaluation Metric	T60	T120	T200
Semantic Coherence using Wordnet (LESK)	0,35	0,23	0,31
Kurtosis (γ_2)	0,49	0,49	0,56
Normalized Maximum Similarity ($z_{\max}$)	0,49	0,50	0,53

ontology, to score topics by measuring the average distance between words of a topic using a variety of metrics based on the ontology. The best performing metric was reported to be the LESK-metric [10], based on lexical overlap in dictionary definitions. Table 3.1 shows the Spearman rank correlation between the LESK score for each topic and the manually assigned quality scores. For comparison, the rank correlation between the manual quality scores and γ_2 and $z_{\max}$ (as calculated in Section 3.3.1) are shown as well, and lead to significantly higher correlation values than with the LESK metric.

3.3.3 Graphical Alignment of Topics

We can visualize the alignment between the supervised and unsupervised topics by calculating the kurtosis on the similarities between both topic sets. These are shown in Fig. 3.2, a *correspondence chart* similar to the one presented in [5], for the 60 topics LDA-model (T60). Our chart differs from the one presented in [5] in that it uses topics from an L-LDA model for the matching of unsupervised topics instead of a list of words generated by experts, and uses bar-charts to display the automatically calculated kurtosis scores instead of likelihoods of human-assisted matching. The size of the circles denotes the cosine similarity between the corresponding supervised and unsupervised topics, and the coloring shows which concepts are matched in a one-to-one fashion by the unsupervised and supervised topics using the harmonic mean of both kurtosis' values. Note that the detection of topics is dependent on the labels included in the supervised data. High-quality LDA-topics, not present in the supervised set, are not detected.

The chart shows that topics involving *Christmas*, *Fire* and *Water* are all very distinguishable by statistical models and human-assisted labeling, or resolved. Other topics are linked to more labels and contain fused concepts or junk. Another use of this chart is evaluating the reference topics by the experts of GOS. Some concepts devised by experts may be chosen too broadly. For example, the supervised topic of *Music/Rocking* is close in cosine-distance to Topic 6 and to Topic 54, which in turn is close to the supervised theme *Dancing/Party*. This indicates that labeling for *Mu-*

sic/Rocking should be confined more to music and exclude songs about dancing. Topics like *Love* and *Heartbreak* correlate with many LDA-topics which demonstrate their dominance in lyrical themes.

3.4 Conclusion

This paper provides insights into the quality of topic models constructed from song lyrics using a small supervised reference. We showed that the kurtosis is a suitable metric to align unsupervised topics with supervised reference topics, which allows detecting high-quality topics in accordance to manual quality assessments.

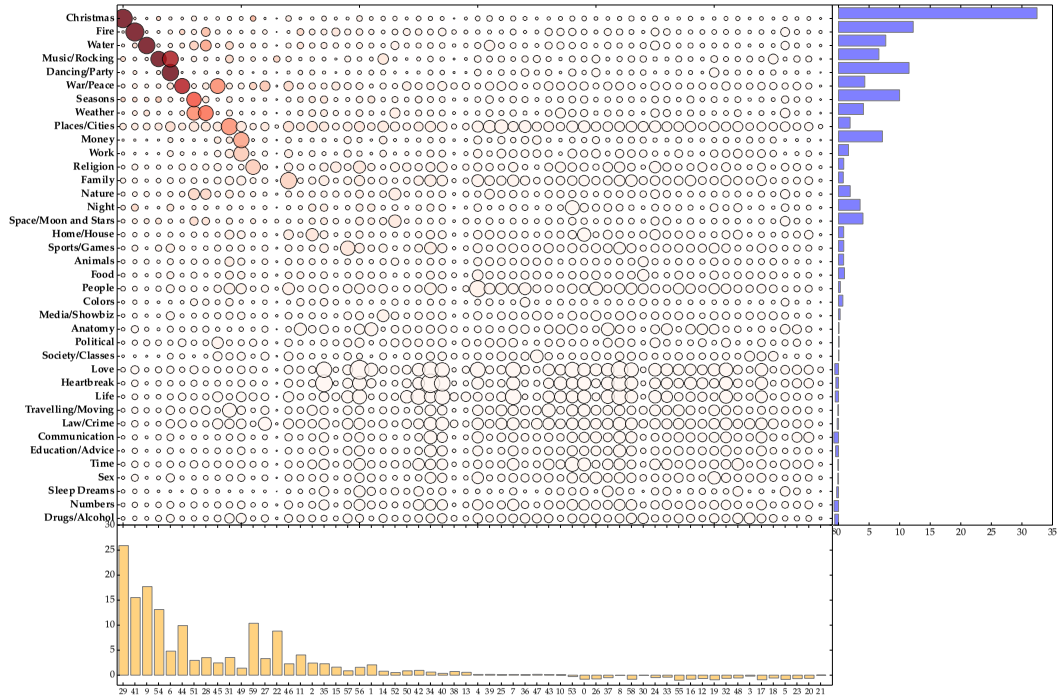


Figure 3.2: Correspondence Chart between topics. The size of circles depicts cosine similarity between corresponding supervised and unsupervised topics. Bars on the sides of the graph show the kurtosis scores for their corresponding topics. High scores show that topics are aligned, low scores mean topics are not matched or junk in the case of LDA-topics. Circle coloring means a strong match between LDA and L-LDA topics, thus a useful unsupervised topic.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. *Latent dirichlet allocation*. The Journal of Machine Learning Research, 3:993–1022, 2003.

[2] B. Logan, A. Kositsky, and P. Moreno. *Semantic analysis of song lyrics*. In International Conference on Multimedia and Expo, ICME 2004., volume 2, pages 827–830. IEEE, 2004.

[3] F. Kleedorfer, P. Knees, and T. Pohle. *Oh oh oh whoah! towards automatic topic detection in song lyrics*. In Proceedings of the 9th ISMIR, pages 287–292, 2008.

[4] D. Newman, S. Karimi, and L. Cavedon. *External evaluation of topic models*. In Australasian Document Computing Symposium (ADCS), pages 1–8, 2009.

[5] J. Chuang, S. Gupta, C. D. Manning, and J. Heer. *Topic Model Diagnostics: Assessing Domain Relevance via Topical Alignment*. In ICML, 2013. Available from: http://vis.stanford.edu/papers/topic-model-diagnostics.

[6] B. McFee, T. Bertin-Mahieux, D. P. Ellis, and G. R. Lanckriet. *The million song dataset challenge*. In Proceedings of the 21st international conference companion on World Wide Web, pages 909–916. ACM, 2012.

[7] A. K. McCallum. *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu, 2002.

[8] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. *Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora*. In Proceedings of the 2009 Conference on EMNLP, pages 248–256. ACL, 2009.

[9] K. V. Mardia. *Measures of multivariate skewness and kurtosis with applications*. Biometrika, 57(3):519–530, 1970.

[10] M. Lesk. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*. In Proceedings of the 5th annual international conference on Systems documentation, pages 24–26. ACM, 1986.

4

Creation and Evaluation of Large Keyphrase Extraction Collections with Multiple Opinions

Automatic keyphrase extraction is the task of automatically extracting the most important and topical phrases of a document. Proper evaluation of keyphrase extraction requires large test collections with multiple opinions which were not available for research. In this paper, we developed large corpora of news, sports and fashion articles annotated with keyphrases by a diverse crowd of laymen and professional writers. Prior, there was little consensus on the definition of the task of keyphrase extraction, with few benchmark collections of keyphrase-labeled data, and a lack of overview of the effectiveness of different techniques. We benchmark existing techniques for supervised and unsupervised keyphrase extraction on the newly introduced corpora. This research was part of the STEAMER project, a collaboration with major Flemish media companies to implement various information retrieval techniques to aid professional writers and funded by IWT (now known as Flanders Innovation & Entrepreneurship) and Innoviris.

⋆ ⋆ ⋆

L. Sterckx, T. Demeester, J. Deleu and C. Develder

Appeared in Language Resources and Evaluation, 2017.

Abstract While several Automatic Keyphrase Extraction (AKE) techniques

have been developed and analyzed, there is little consensus on the definition of the task and a lack of overview of the effectiveness of different techniques. Proper evaluation of keyphrase extraction requires large test collections with multiple opinions, currently not available for research. In this paper, we (i) present a set of test collections derived from various sources with multiple annotations (which we also refer to as *opinions* in the remainder of the paper) for each document, (ii) systematically evaluate keyphrase extraction using several supervised and unsupervised AKE techniques, (iii) and experimentally analyze the effects of disagreement on AKE evaluation. Our newly created set of test collections spans different types of topical content from general news and magazines, and is annotated with multiple annotations per article by a large annotator panel. Our annotator study shows that for a given document there seems to be a large disagreement on the preferred keyphrases, suggesting the need for multiple opinions per document. A first systematic evaluation of ranking and classification of keyphrases using both unsupervised and supervised AKE techniques on the test collections shows a superior effectiveness of supervised models, even for a low annotation effort and with basic positional and frequency features, and highlights the importance of a suitable keyphrase candidate generation approach. We also study the influence of multiple opinions, training data and document length on evaluation of keyphrase extraction. Our new test collection for keyphrase extraction is one of the largest of its kind and will be made available to stimulate future work to improve reliable evaluation of new keyphrase extractors.

4.1 Introduction

Automatic keyphrase extraction (AKE) is the task of automatically extracting the most important and topical phrases of a document [1]. Keyphrases are meant to cover all topics and capture the complete content of a document in but a handful of phrases. Applications of keyphrases are rich and diverse, ranging from document summarization [2] to clustering [3], contextual advertisement [4], or simply to enhance navigation through large corpora. While much research has been done on developing supervised [5–8] and unsupervised methods [9–12], scores for recall and precision for this task are well below those of standard NLP tasks such as POS-tagging or Named Entity Recognition. This is due to a variety of difficulties faced when extracting keyphrases, including the inherent ambiguity of the task, flaws in evaluation measures (e.g., semantically identical keyphrases are judged as different), the over-generation of keyphrases, etc. One of the most pressing issues in AKE research is the lack of large test collections with multiple opinions. In this paper we aim to address that gap and thus provide initial answers to the still open questions in solving and evaluating AKE, e.g., *What is the agreement on keyphrases among multiple readers? How*

well do the keyphrase candidates generated using the standard candidate generation procedures match keyphrases assigned by annotators? How do supervised and unsupervised methods compare?

Keyphrase extraction has a long history, used in libraries for archiving and cataloging purposes. In such a library setting, keyphrases are assigned by trained experts using detailed manuals and rules, such as the Anglo-American Cataloging Rules (AACR) in Encyclopedia of Library and Information Sciences [13], or the German libraries' "Regeln für den Schlagwortkatalog (RSWK)".[1] However, the setting we consider in our work concerns AKE for popular media articles, where keyphrases will be used by a typically untrained (layman) audience. Thus, also annotators will be laymen, and the keyphrase setting is therefore less constrained and the phrase importance fairly open to interpretation. The key contribution of this paper is the creation of a new dataset (4 corpora of 1000-2000 documents each) and a comparison in performance of common supervised and unsupervised AKE strategies.

First, in Section 4.2, we describe the construction of our new set of large and diverse collections of documents annotated with keyphrases. Next, Section 4.3 gives an overview of common AKE techniques and presents several strategies to include context-dependent features into supervised models, leading to increased precision. In Section 4.4, we evaluate the performance of the presented supervised and unsupervised AKE techniques, which apply knowledge extracted from background corpora (e.g., under the form of topic models). We study the influence of the amount of training data on AKE performance and point out the relatively low annotation effort needed to train competitive supervised models. In Section 4.5 we conclude by providing readers with guidelines to keep in mind when researching AKE and evaluating new techniques.

4.2 Test Collections

The state-of-the-art in AKE is not only diverse in terms of techniques (see further, Section 4.3), but also in terms of test collections used for evaluation. Indeed, these vary from formal scientific articles [14] to more popular content such as mainstream news, or even blogs and tweets [15]. Issues with these evaluations are that (i) most collections are fairly limited in size (typically a few hundred documents) and (ii) annotations substantially vary from one collection to the next, since they are performed by either the various authors of the content or a single reader assigning keyphrases to many different documents, with possibly different annotation guidelines or goals from one collection to the next. As [16] noted, "for scientific articles the authors do not always choose keyphrases that best describe the content of

[1] http://www.dnb.de/DE/Erwerbung/Inhaltserschliessung/rswk.html

their paper, but they may choose phrases to slant their work a certain way, or to maximize its chance of being noticed by searchers." Due to these limitations, and with the existing collections for AKE, it is hard to study how AKE performance may be impacted by annotators and the type or topic of documents.

We set out to systematically construct a rich and diverse set of annotated test collections to investigate this issue. We particularly focus on rather popular content targeted to a diverse, layman audience (e.g., as opposed to specialist scientific literature). Our newly created set of test collections (i) is substantial in size with four different collections of 1200–2000 annotated documents each, (ii) comprises different types of news content (online news, online sports, lifestyle magazines, newspaper articles), and (iii) has each document annotated by multiple annotators (on average 6 per document), where annotator guidelines are identical for all collections (i.e., annotators are only informed with the definition and purpose of the keyphrases, regardless of the collection the documents are drawn from).

These collections are available for research purposes.[2]

4.2.1 Document Collection

In order to procure the test collections, we started from a large collection of candidate documents provided by three major Belgian media companies—each with their own distinct type of content (all in Dutch). The first media company involved, is the public-service broadcaster VRT, who offered two collections: *Online News* and *Online Sports*. The *Online News* collection is a subset of the texts accompanying the videos on its official news channel website De Redactie.[3] Similarly, the *Online Sports* collection represents their specialized sports section Sporza.[4] The second company, Sanoma, is a publishing group owning a selection of lifestyle, fashion, and health magazines, from which we created the *Lifestyle Magazines* test collection. The third company, Belga[5], offers a digital press database comprising content from all Flemish and Dutch newspapers, represented in our *Printed Press* set.

To verify that these collections indeed contain different topics, we use an external multi-label document classifier[6] trained on documents annotated with IPTC media codes[7] to gain insight into the thematic subjects

[2]For information regarding acquiring the test collections, please contact the paper's first author.

[3]http://www.deredactie.be

[4]http://www.sporza.be

[5]http://www.belga.be

[6]Our multi-label classifier is based on methods from top submissions in the "Greek Media Monitoring Multilabel Classification" (https://www.kaggle.com/c/wise-2014) and "Large Scale Hierarchical Text Classification" (https://www.kaggle.com/c/lshtc) hosted by Kaggle.

[7]https://iptc.org/standards/media-topics/

covered. The average contributions of IPTC codes (at the first of three levels in the IPTC codes) are shown in Table 4.1, and confirm our intuition as humans being familiar with the various document collections: a large focus on sports texts for the *Online Sports* collection, mostly political subjects in *Online News*, lifestyle topics in *Lifestyle Magazines* and general news (dominated by sports) in *Printed Press*.

4.2.2 Collecting Keyphrases

Documents were presented to a panel of 357 annotators of various ages and backgrounds (selected and managed by imec.livinglabs[8]), who were asked to "*select a limited number of short phrases that concisely summarize the document's contents*". Three annotation sessions, of each spanning two weeks, were organized. Annotators were allowed to participate in multiple sessions. For each session an annotator was assigned 140 articles but was not obligated to finish the complete assignment. Compensations were awarded at 60, 100 and 140 articles. The amount of documents annotated by each of the 357 annotators is shown in Fig. 4.1 a. To ensure overlap, each document was included in ten different annotators' task lists. Depending on the annotators' effort each document received at least one and up to ten different opinions. The final distribution of overlap per document is shown in Fig. 4.1 b. Overall, 26% of the documents received more than 8 opinions. Other descriptive statistics on the annotator panel are shown in Tables 4.2 and 4.1. As briefly mentioned in the Introduction, our annotation setup is quite different from traditional keyphrase annotation scenarios, where typically a small number of well-trained assessors provide annotations according to strict rules. In our setting, there was a large number of annotators, sampled from the Flemish media audience, which forms the target group for applications built on the extracted keyphrases. Also, we did not impose strict annotation rules, and instead propose that the results reflect the expectation of what keyphrases should look like for the target audience. The simplicity of our setup could be an important advantage for organizations intending to build a keyphrase extraction system on their own data. However, the lack of strict rules also implies potential issues of disagreement on the chosen keyphrases among different annotators (see Section 4.2.5).

4.2.3 Annotation Tool

A web application was built for the test panel to perform annotations using a web browser from home. The annotation process works as follows. Annotators log in to the application using a personalized password. Each an-

[8]https://www.iminds.be/en/succeed-with-digital-research/go-to-market-testing/proeftuinonderzoek

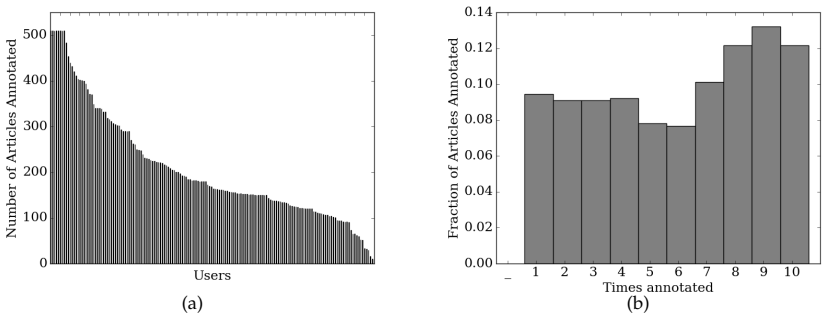


Figure 4.1: (a) Amount of annotated documents per annotator. (b) Distribution of overlap per document.

notator was then directed to a briefing on the meaning and purpose of key-phrases and how to use the application to enrich articles with key-phrases. These guidelines are explained in detail in the following Section. The application chooses an article from the total collection stored in a database and presents it to the annotator. Articles are selected to increase overlap of annotators per document as fast as possible. A first document gets ranked first in the task list of 10 different annotators, a second document is then ranked second for these ten annotators. This is repeated until each annotator received a task list of 140 articles to be annotated during the two week session. Keyphrases are selected by sliding the mouse pointer over a selection of words. Unlike many keyphrase extraction tasks where authors assign free-form phrases to a document, this means keyphrases are guaranteed to appear in the text of the document. A theoretical upper bound for an extractor solely from the text thus would be 100%. One of the reasons this is often imposed, is the intended use of the keyphrases to highlight the most important phrases in the articles themselves. While this confines the task as certain key concepts do not explicitly appear in the text as is done in *in-line keyphrase extraction* [**?**, 14], extraction of such keyphrases is a problem requiring different strategies, thus, we should not penalize an *inline* keyphrase extractor for not extracting these keyphrases. Documents are tokenized before annotation and annotators' highlighting is confined to token boundaries to facilitate annotating and reduce matching errors afterwards. The annotator is then prompted to add the selection as keyphrase, after which the keyphrase is shown in a list next to the article with other assigned keyphrases.

All keyphrases are highlighted after selection. Figure 4.2 shows the application as displayed in the web browser. The annotator is also provided with a button to send a form to provide feedback on a specific article, e.g., to indicate confusing cases or articles not suitable for annotation (such as

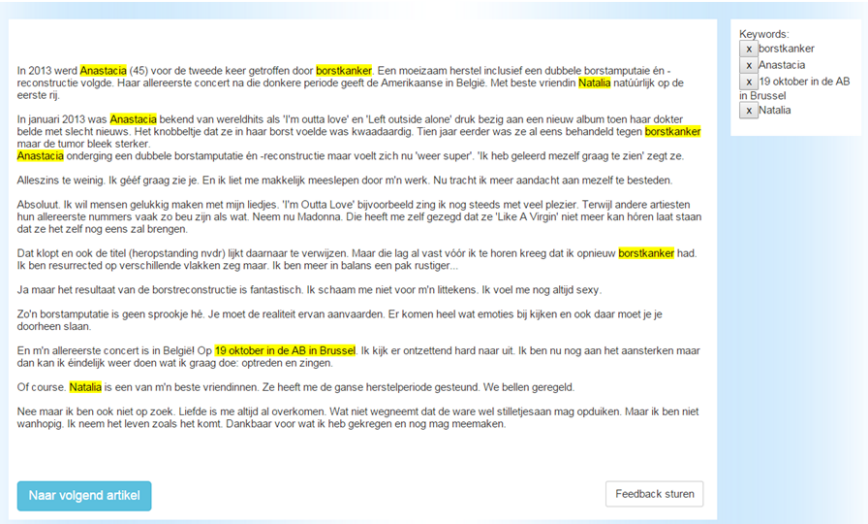


Figure 4.2: Web interface for annotation of keyphrases.

articles featuring two entirely different topics or stories, tables of sports results, cooking recipes, etc.).

4.2.4 Keyphrase Annotation Guidelines

The full guidelines section of the annotation tool is shown in Fig. 4.3. In addition to guidelines and videos, annotators were provided with multiple examples of annotated documents. We keep to the standard definition of keyphrases [1] and impose no constraints on keyphrase form. Whereas the annotation procedure differs from one collection in literature to the next, all of the currently presented test collections were created the same way. No prior limits or constraints were set on the amount, length, or form of the keyphrases. This allows us to study the interpretation by annotators and their disagreement, as well as investigate the form of typical keyphrases that candidate generation approaches should produce. Fig. 4.4 shows an example of a (translated) Dutch lifestyle article about music artist *Anastacia* and her recovery from breast cancer, where bold text represents keyphrases as annotated by 10 different annotators. A superscript of i indicates the identifier of the annotator who selected this phrase as keyphrase. This example is a first demonstration of the lack of consensus on keyphrases: only few keyphrases are selected by all of the annotators. We expand on this disagreement issue in Section 4.2.5.

In Table 4.1 we present descriptive statistics on the length of the documents, the amount of assigned keyphrases, the amount of keyphrase candidates per document (candidate generation is presented in the following

Dear Annotator,
Thank you for participating in the Steamer Bootcamp! The next 2, 4 or 6 weeks, you will read a lot of news reports and select the most important keywords or keyphrases in them.

Your aid is of major importance
Depending on your choice of keywords, we will develop a system that automatically recognizes these keywords and adds them to documents. These keywords are not only very useful for many applications, they also make it easy for search engines to improve the automatic recommendation of other relevant articles -for you-.

Keyphrases?
The keywords or 'keyphrases' are defined as "a selection of short, significant expressions consisting of one or more words that can summarize the article very compactly."

Too complicated?
Below you find some videos back with a quick guide.
<Link to instruction video> Reading articles.
<Link to instruction video> Indicating keywords.

Method
There are some tips to get to the best keywords:
Ask yourself, "What words summarize the content of the article?" Or "What words are most representative of the contents of the article?" This can be an event or an object, the crucial entities, or organizations that are mentioned in the article. Try to keep the keyphrase as short as possible. Words that do not contribute may be omitted to the meaning of the keyphrases. The number of keywords per article depends largely on the length of the article and the various topics discussed in it. It is rare to select more than 10 keywords per article.

We demonstrate this with an example:
"Higher education is bracing itself. Once it had ample offer, now it calls for a economization of supply. The Flemish coalition agrees that the universities themselves must make proposals to achieve a constrained and transparent offer. In interviews, the Minister of Education, Hilde Crevits (ISA), indicates that the offer can be safely pruned to one hundred of majors. "
in this example "higher education" , "economization of supply" and " Hilde Crevits" would be appropriate keywords.

Still not clear?
Click <Link to more examples> for more examples.

Select keywords or keyphrases
You select a keyphrase by clicking a phrase in its first word and sliding to the last word in the keyphrase. The tool will then ask if the selected keyword should be added. Afterwards all selected keywords are displayed in the left column. If you've changed your mind, then a chosen keyword can be removed by clicking on the red cross. Think you have selected all the keywords? Save the article and go to the next article.

Ready?
Then you can begin! The more articles you read, the faster you will start to find the keywords. It's a little hard at first, but hang in there, it gets better. Click "Start" and you can start!

Any questions?
Take a look at our FAQ page or use the feedback button. Good luck!
Greetings,
The Steamer research team

Figure 4.3: Instructions shown at the main page of the keyphrase annotation tool.

In 2013 **Anastacia**[1,2,3,4,5,6,7,8,9,10](45) was struck for the second time with **breast cancer**[1,2,3,4,5,6,7,8,9,10]. A difficult recovery, including a double **mastectomy**[4] and reconstruction followed. She gave her **first concert**[2] in Belgium after this dark period, with her best friend **Natalia**[3,4,8,9,10] in the front row. In January 2013 Anastacia was known for world hits like '**I'm outta love**'[3] and '**Left Outside Alone**'[3]. Busy working on a new album, her doctor called with bad news. The lump she felt in her breast was cancerous. Ten years earlier she had already been treated for breast cancer, but the tumor appeared stronger. Anastacia underwent a **double mastectomy**[10] and reconstruction, but now feels "great again". "I've learned love to see myself," she says. "I am happy, you see. And I let myself be carried away by my work easily. Now I try to pay more attention to myself. Absolutely. I want to make people happy with my songs. 'I'm Outta Love', for example, I still sing that with pleasure. While other artists are often tired of their first songs. Take Madonna. Who said herself that she can't hear 'Like A Virgin', let alone sing it herself. That's right, and even the title ('**Resurrection**[7]', ed) seems to refer to it. But this was already determined before I was told I had cancer again. I am resurrected in different areas, so to speak. I'm more balanced, a lot calmer ... Yes, but the result of the **breast reconstruction**[5] is fantastic. I'm not ashamed of my scars. I still feel sexy. Such a **mastectomy**[8] is not a fairy tale, huh. You must accept the reality. There are a lot of emotions involved, and you have to beat you through. And my **first concert**[3,10] is in **Belgium**[2] On October 19, in the AB in Brussels I can finally do what I love to do: act and sing. Of course, Natalia is one of my best friends. She has supported me the whole recovery period. We call regularly. No, but I'm not searching. Love has always happened to me. Which does not mean that true love may turn up little by little. But I'm not desperate. I take life as it comes and I am grateful for what I've got."

Figure 4.4: Example of annotated article, with indication of keyphrase annotations by 10 different annotators using superscripts.

section), the entities per document, the distribution over n-grams in keyphrases, predicted topics and POS-tags, for the four different test collections. As Table 4.1 indicates, the largest difference between the test collections is their thematic content. Articles from the collections are relatively short, with *Printed Press* featuring slightly longer articles than the three other collections. *Online Sports* articles contain more entities, with notably more entities that are seen as keyphrase by the annotators. On average, a single annotator assigns 5 keyphrases to each document. The union of all annotations per document on average contains 15 keyphrases.

4.2.5 Annotator Disagreement

Multiple annotations for each document show that the notion of "*what is a keyphrase?*" remains subjective. Figure 4.5 shows the fraction of annotated keyphrases for different ratios of overlap by the complete set of annotators. This shows that the largest fraction of all keyphrases ($\geq 50\%$) are selected by less than 20% of all the annotators that assigned keyphrases to the document. This is due to different interpretations of the article, but also due to keyphrases with equal semantics appearing in different forms. This has important consequences for training models on keyphrases annotated by a single annotator: in such a setting, many alternate candidate

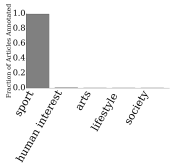
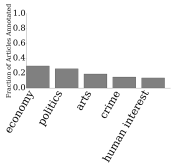
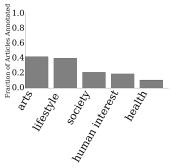
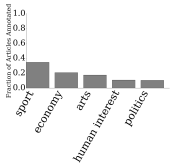
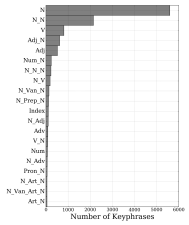
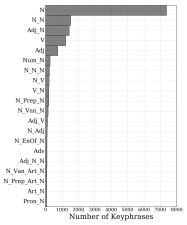
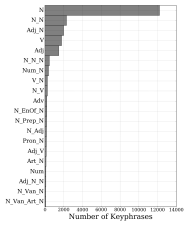
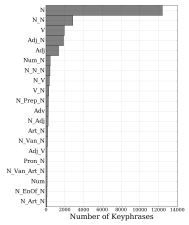
	Test Collections			
Name	Online Sports	Online News	Lifestyle Magazines	Printed Press
Type (date range generated content)	Sports Articles (2012-2014)	Television News (2012-2014)	Fashion, Lifestyle… (2010-2013)	News Articles (2009-2014)
# Documents	1,252	1,259	2,202	2,196
# Keyphrases	14,544	19,340	29,970	31,461
⊘ Keyphrases/Annotator	4.6	5.7	4.7	4.8
⊘ Keyphrases/Document	11.6	15.4	13.7	14.4
⊘ Tokens/Keyphrase	2.1	2.3	2.0	1.8
⊘ Tokens/Document	288	332	284	399
⊘ Tokens/Candidate phrase	1.8	2.0	1.9	1.5
⊘ Candidate Keyphrases/Doc.	43	52	49	67
⊘ Entities/Document	10.6	5.7	4.1	8.3
⊘ Entities/Keyphrases (%)	30.6	12.7	13.4	18.1
1/2/3/3+ -gram distribution (%)	52 / 35 / 8 / 5	55 / 27 / 9 / 9	58 / 25 / 9 / 8	57 / 28 / 8 / 7
Max. POS-filter Recall (%)	65.9%	65.5%	59.5%	64.2%
IPTC Theme Distribution				
Distribution of Keyphrase POS-Tags				

Table 4.1: Corpus statistics for the four annotated keyphrase datasets used in this paper. We describe the amount of documents and keyphrases, average length of the documents and the keyphrases, the tokens in the keyphrases, the average amount of keyphrases assigned to documents, the distribution of keyphrases over n-grams, total and average amount of entities present in keyphrases. Plots show the distribution of the topics detected in the collection by a multi-label classifier and the distribution of POS-tag sequences of keyphrases.

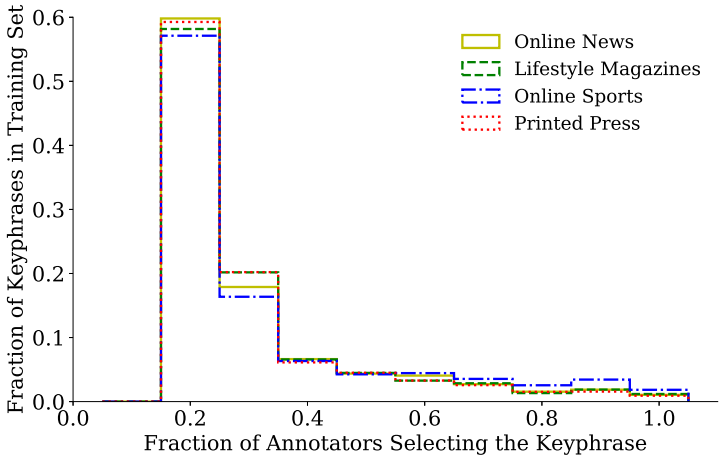


Figure 4.5: Illustration of annotator disagreement on keyphrases. The X-axis shows the fraction of annotators that agree on selecting a single keyphrase for a given document. For example, if we were to restrict keyphrases to those selected by 50% of the annotators, this shows that we would retain less than 5% of all keyphrases.

Table 4.2: Descriptive statistics regarding the annotations (# = number of; ⌀ = average).

# Annotators	357
# Documents with $\geq$ 1 annotation	7342
max. Annotators/Document	10
min. Annotators/Document	1
⌀ Annotators/Document	6
⌀ Articles/Annotators	140

phrases that other annotators would pick, would be considered as negative training data. The performance on evaluation sets can thus greatly vary depending on the annotator of the test set. Studying disagreement and the effect of training data by different annotators on the evaluation confidence is a valuable direction for future research. In our evaluation of automatic extractors, we use the aggregated set of keyphrases as a reference but also report on the average and standard deviation on scores for different reference keyphrase sets assigned by different annotators.

As metric for inter-annotator disagreement we report the Fleiss' kappa score [17]. We define a Fleiss' kappa, κ_F, for each document as

$$\kappa_F = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}. \tag{4.1}$$

Here, $1 - \bar{P}_e$ measures the degree of agreement that is attainable above chance, and, $\bar{P} - \bar{P}_e$ measures the degree of agreement achieved above chance. For these formulas, we consider the N generated keyphrase candidates as rated items, that are scored by each of the n annotators with one of $k = 2$ possible scores, to represent the cases where a given phrase is annotated as a keyphrase or a non-keyphrase by a given annotator. To find $\bar{P}$ and $\bar{P}_e$, first p_j, the proportion of all assignments as keyphrase ($j = 1$) or non-keyphrase ($j = 2$), is calculated:

$$p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij}. \tag{4.2}$$

Then P_i is calculated, which is extent to which annotators agree on the i-th keyphrase candidate.

$$P_i = \frac{1}{n(n-1)} \left[\left(\sum_{j=1}^{k} n_{ij}^2 \right) - n \right]. \tag{4.3}$$

$\bar{P}$ is the mean of the P_i's and $\bar{P}_e$ is the sum of the squared p_j's, which are then used to calculate κ_F.

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i, \quad \text{and} \quad \bar{P}_e = \sum_{j=1}^{k} p_j^2. \tag{4.4}$$

Median Fleiss kappa across all documents is low at 0.19 with slightly higher values for *Online Sports'* articles. As presented in Table 4.3, higher agreement for sports articles might be due to entities being central to many of these articles. The annotators were recruited from among the target audience in the Flemish media landscape by experienced Living Lab researchers [18], including both residential consumers and media professionals. For the experiments presented in this work, we pooled all annotators. However, the dataset contains additional information that would

Table 4.3: Annotator agreement on keyphrases, quantified with Fleiss' kappa κ_F, is quite low.

Collection	Online Sports	Online News	Lifestyle Magazines	Printed Press
Avg. κ_F per doc.	0.235	0.189	0.193	0.186

allow making a distinction between different types of annotators, in order to study different use cases. For example, we quantified the difference in annotation behavior for the most active half versus the least active half of the annotators. The 50% most active annotators are responsible for 82% of all sets of keyphrases, assigning on average 4.1 keyphrases per document, each on average 2.2 tokens long. The other half assigned on average 5.2 keyphrases per document, but slightly longer ones, with on average 2.9 tokens.

4.3 Keyphrase Extraction Techniques

This section provides a brief overview of common AKE techniques. After explaining how candidate keyphrases are selected (Section 4.3.1), and how well these common heuristics cover the annotated keyphrases, the most prominent unsupervised methods are introduced (Section 4.3.2). Next, supervised keyphrase extractors and feature design for AKE are presented (Section 4.3.3).

4.3.1 Candidate Selection

To avoid spurious keyphrase instances, and to limit the number of candidates, extractors choose a subset of phrases which are selected as candidate keyphrases. Especially for long documents, the resulting list of candidates can be long and hard to rank. Current state-of-the-art mainly adopts part-of-speech (POS) filters, typically after stopword removal. Other heuristics for selecting candidates only allow keyphrases from a curated, fixed list [19] or Wikipedia article titles [20], which drastically reduces the amount of possible candidates. Here, we quantitatively address the open question as to what extent such POS-filtered keyphrases correspond to those assigned by human annotators. For that purpose, we calculated the measure "Maximum POS-filter Recall" shown in Table 4.1 for each collection, defined as the recall attained by the most common POS-filter based on the rules defined in [7] over the set of human annotator keyphrases. More

specifically, the filter is defined by the following regular expression[9]:

$$(<\text{Adj}|\text{Num}>^* <\text{N}>+<\text{IN}|\text{Van }>)?<\text{Adj}|\text{Num}>^*<\text{N}>+ \quad (4.5)$$

Applying this common filter to our data sets shows that, if we were to use it to select candidate phrases from the text, we would maximally reach a recall of about 66% when considering *all* the annotated keyphrases as gold standard, as listed in Table 4.1. This relatively low coverage demonstrates the mismatch between POS-filters and the interpretation of the keyphrase concept by the (layman) annotators. POS-filters also extract longest matching sequences of tokens, while keyphrases might be subsequences. Figure 4.6 shows the complete distribution of POS patterns assigned to the annotated keyphrases versus those of extracted candidates by the POS-tagger. While the majority of keyphrases are Noun Phrases, it is shown that a considerable fraction of keyphrases are not extracted by the standard POS-filter, such as lone verbs and adjectives. This indicates that people also tend to see actions or events, denoted by a verb, central to an article's content. A topic for future research is to maximize the coverage of keyphrases by candidates while limiting the total amount of extracted candidates, i.e., , the trade-off between recall (as the maximum achievable amount increases), and precision (as keyphrase extraction becomes harder with more candidates to rank correctly). What POS-filter is the most effective (for optimal recall versus precision) also depends on the type of document: while news articles describe events, typically requiring entities and verbs as keyphrases, we expect this to be less the case for scientific articles where domain specific technological terms are more common. We advise to adapt the candidate generation procedure appropriately.

4.3.2 Unsupervised Keyphrase Extraction

A disadvantage of supervised approaches is the requirement of training data and the resulting bias towards the domain of the training data, undermining their ability to generalize well to new, unseen domains. This limitation is bypassed by unsupervised approaches that focus on word-frequency or centrality in graph transformations [10, 21–23]. Note that unsupervised approaches have been reported as state-of-the-art on many test collections [12]. Because most of these test collections do not supply a training and test data split, comparisons of these models with supervised models is missing. The most important unsupervised AKE approaches are (variations on) the following baseline methods:

- **TF*IDF** [24] is the most common strategy for ranking keyphrases for a given document in both unsupervised and supervised models [20, 25]. The TF*IDF weight consists of two factors: TF is the

[9] POS-tag definitions used here: Adj = adjective, N = nouns (including singular and plural), IN, Van = preposition or subordinating conjunction and Num = quantity expressions.

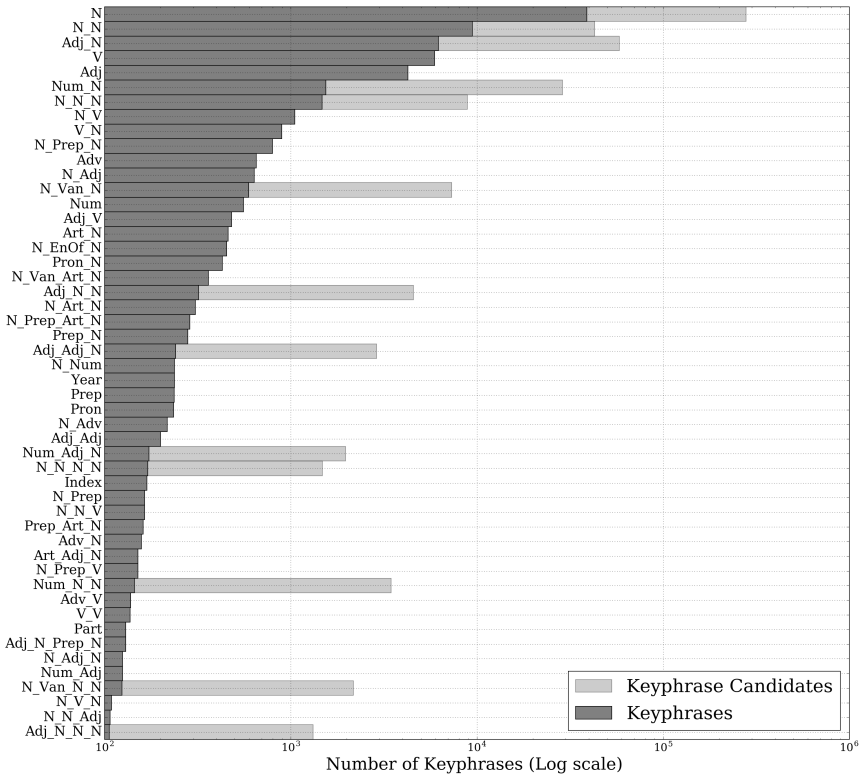


Figure 4.6: POS-tags of extracted keyphrase candidates by filters versus complete distribution of all the annotated keyphrases from all collections.

frequency of the considered keyphrase. The second factor, IDF, is the Inverse Document Frequency, computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific phrase appears. The IDF factor is incorporated to diminish the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

- **TextRank** is a completely within-document AKE technique [26], which represents the document as a graph. Each word corresponds to a node in the graph, edges are created between words co-occurring within a window of pre-defined width. Centrality of the nodes is then calculated using PageRank. Keyphrases are generated using high-scoring nodes and by merging co-occurring terms. In our evaluation we use the **SingleRank** variant [11], in which edges are weighted according to the number of times they co-occur within the window.

The score for word w_i is computed iteratively until convergence using the recursive formula:

$$S(w_i) = \lambda \cdot \sum_{j:w_j \to w_i} \left(\frac{e(w_j, w_i)}{O(w_j)} \cdot S(w_j) \right) + (1 - \lambda) \tag{4.6}$$

where $S(w_i)$ is the PageRank score for word w_i, $e(w_j, w_i)$ is the weight of the edge $(w_j \to w_i)$, the number of outbound edges is $O(w_j) = \sum_{w'} e(w_j, w')$ and λ is a damping factor $\in [0, 1]$ indicating the probability of a random jump to another node in the word graph.

- **Topical PageRank**, as described in [10], calculates a PageRank score separately for each topic in a pre-trained topic model and boosts the words with high relevance to the corresponding topic. That topic-specific PageRank score for word w_i is defined as follows:

$$S_z(w_i) = \lambda \cdot \sum_{j:w_j \to w_i} \left(\frac{e(w_j, w_i)}{O(w_j)} \cdot S_z(w_j) \right) + (1 - \lambda) \cdot P_z(w_i), \tag{4.7}$$

where $S_z(w_i)$ is the PageRank score for word w_i in topic z. A large $S_z(w_i)$ indicates that w_i is a good candidate keyword in topic z. The topic specific preference value $P_z(w_i)$ for each word w_i is the probability of arriving at this node after a random jump, thus with the constraint $\sum_{w \in v} P_z(w) = 1$ given topic z. In TPR, the best performing value for $P_z(w_i)$ is reported as being the probability that word w_i occurs given topic z, denoted as $P(w_i|z)$. This indicates how much that topic z is focused on word w_i. With the probability of topic z for document d $P(z|d)$, the final ranking score of word w_i in document d is computed as the expected PageRank score over that topic distribution, for a topic model with K topics,

$$S(w_i) = \sum_{z=1}^{K} S_z(w_i) \cdot P(z|d). \tag{4.8}$$

We apply the more efficient, equally effective, single-PageRank variant proposed in [22]. Other graph based methods using background information are based on relatedness between candidates in the document in thesauri [27].

4.3.3 Supervised Keyphrase Extraction

Supervised methods recast the extraction problem as a binary classification task, where a model is trained to decide whether a candidate phrase (generated from the candidate generation procedure discussed in Section 4.3.1)

is a keyphrase or not [1, 5, 16]. Treating automatic keyphrase extraction as a supervised machine learning task means that a classifier is trained using documents with known keyphrases. While the decision is binary, a ranking of phrases can be obtained using classifier confidence estimates, or alternatively, by applying a learning-to-rank approach [28].

4.3.3.1 Feature Design and Classification

An important aspect of supervised approaches is feature design. In previous work, many features have been designed and reported as being effective on different occasions [1, 5–8, 29]. In these studies, several types of features can be distinguished:

- **Statistical Features:** Features such as the term frequency, TF*IDF (discussed in the following section) and keyphraseness (the total amount of times a keyphrases occurs in a training collection).

- **Structural Features:** Features characterizing the position of a term with respect to the document structure (first location, last location, occurrence in title, etc.),

- **Content:** Features characterizing the keyphrase, such as the lexical cohesion of the keyphrase (Dice Coefficient of the tokens in the keyphrase and the complete keyphrase), length, the POS-pattern, capitalization, etc.

- **External Resource Based Features:** information is added using external resources or dictionaries such as terminological resources (Medial Subject Headings (MeSH), the Gene Ontology, etc.), linguistic resources (WordNet), thesauri [30], Wikipedia, topic models, or tags from a Named Entity Recognizer.

After features are extracted, a learning algorithm is applied on the training collection to distinguish keyphrases from non-keyphrases. Many different statistical classifiers have been applied for this task, including Naive Bayes [31], bagging [5], max-entropy [4], multilayered perceptron [6],support vector machine [28] and (boosted) decision trees [6]. A detailed comparison with each of the designed features in existing supervised techniques is not in the scope of this paper. As for many supervised machine learning tasks, a classifier needs to be developed, evaluated and separately optimized for each collection (also known as the *no-free-lunch theorem* [32]). We propose a different approach and develop a baseline supervised extractor, compare with unsupervised techniques, propose several features modeling the context of the document, and study the influence of the background collection on the AKE effectiveness.

4.3.3.2 Supervised Model

As baseline supervised keyphrase extractor, we extract a number of features based on prior work presented in the previous section. Features effective during development and used in the baseline model are: (i) keyphrase frequency, (ii) number of tokens in the keyphrase, (iii) length of the longest term in the keyphrase, (iv) a binary feature which indicates whether the keyphrase contains a named entity, (v) relative position of the keyphrase's first occurrence in the full article, (vi) relative position of last occurrence and (vii) span (relative last occurrence minus relative first occurrence).

Apart from features extracted from the document the keyphrase appears in, we calculate two features based on background corpora: TF*IDF ($f_{\text{TF*IDF}}$) and Topical Word Importance (f_{LDA}), which is based on context.

TF*IDF consists of the multiplication two factors: TF is the frequency of the considered keyphrase relative to the document length. The second factor IDF is the Inverse Document Frequency, computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific phrase appears. Topical word importance was introduced in [22] and is based on topic modeling, for which we use standard Latent Dirichlet Allocation [33]. Topical word importance is the similarity between the word-topic probability and the topic-probability from a topic model trained on the background corpora. This similarity, as a feature for a word-document pair (w, d), is determined as the cosine similarity between the vector of topical word probabilities $\mathbf{P}(w|Z) = [P(w|z_1), \ldots, P(w|z_K)]$ and the document topic probabilities, $\mathbf{P}(Z|d) = [P(z_1|d), \ldots, P(z_K|d)]$:

$$f_{\text{LDA}}(w, d) = \frac{\mathbf{P}(w|Z) \cdot \mathbf{P}(Z|d)}{\|\mathbf{P}(w|Z)\| \cdot \|\mathbf{P}(Z|d)\|}. \tag{4.9}$$

This score is usually included as a weight in a biased graph-ranking algorithm, here we also include it as contextual feature. Both context dependent features stem from unsupervised techniques. Features established in literature, but which were found to be ineffective include *keyphraseness* and *keyphrase cohesion* [6]. In all experiments, we use a support vector machine (SVM) classifier with a linear kernel from the *libsvm* library [34] and gradient boosted decision trees implemented in the *XGBoost* package [35]. For token-based features we use the sum and average of the TF*IDF and Topical Word Importance values of the tokens constituting the keyphrase. IDF values and topic models are trained on large background collections stemming from the same source and on a more *general* Wikipedia corpus.

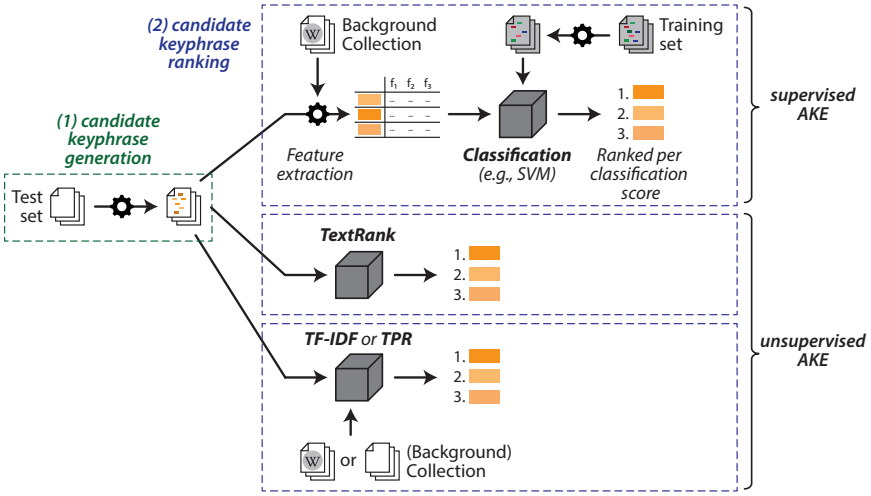


Figure 4.7: Schematic representation of the experimental set-up.

4.4 Systematic Evaluation

In this section, we describe our experimental set-up, followed by an evaluation of different unsupervised and supervised models, with a careful study of the effect of multiple opinions during evaluation and training.

4.4.1 Experimental set-up

Creating training and test set

In Fig. 4.1 b, we showed the amounts of annotations per document. Before separating documents into train and test collections, we remove documents annotated by less than five annotators. From documents with more than five opinions we randomly select five opinions as reference annotations. This is to avoid bias towards more frequently annotated documents, while keeping a reasonable amount of documents to produce meaningful results. From these filtered collections of documents, we sampled 500 documents as test collection for each of the sub-collections, the remaining ones were used for training and development. The amounts of training and test documents are shown in Table 4.4. The amount of training documents ranges from around three hundred to close to a thousand for the different sub-collections, but as will be shown in Section 4.4.5, the effectiveness of the studied supervised approaches saturates beyond a few hundreds of training documents. All annotated documents for the different sub-collections, with listings of document IDs in the training and test

Table 4.4: Number of documents in training and test collections after filtering documents having fewer than five opinions.

	Online Sports	Online News	Lifestyle Magazines	Printed Press
#Background Documents	325,438	325,437	976,318	976,316
#Training Documents	312	275	981	957
#Test Documents	500	500	500	500

collections, as well as extracted candidate keyphrases are made available upon request for research purposes.[10]

System architecture

Figure 4.7 outlines the experimental set-up. After filtering the annotated data and separating train and test set, all documents were further preprocessed by extracting POS-tags using the rule-based Fast Brill Tagger [36], implemented in the NLTK [37] package trained on the CONLL-2002 training set (accuracy of about 95% on the CONLL test sets). Keyphrase candidates were extracted using the POS-filter presented in Section 4.3.1.

For the supervised models, features for each of the candidate phrases and keyphrases (as detailed in Section 4.3) were calculated, and the models were trained on the training subsets.

Contextual information (i.e., IDFs, 1,000-topic LDA-models) was derived for each of the collections individually from a non-annotated background corpus provided by the corresponding media company (ranging from 325,000 documents (*Online Sports, Online News*) to 976,000 (*Printed Press, Lifestyle Magazines*)), as well as from a more universal background corpus, i.e., a 2014 Dutch Wikipedia dump. The Dutch Wikipedia corpus contains 1,691,421 articles, amounting to a total of 226,080,236 tokens.

When processing the documents in the test collection, each AKE approach provides a confidence score to the phrases generated during the candidate generation procedure. For unsupervised models this is the TF*IDF score or PageRank score, whereas for supervised models this is the predicted score, i.e., probability of the candidate being a keyphrase. The candidate keyphrases of the test documents are ranked according to these scores (shown in orange in Fig. 4.7). For converting predicted scores to binary decisions on whether or not to retain the keyphrases, the cut-off scores with highest F_1 score on the training set are used (also for the unsupervised approaches).

Hyperparameter tuning was kept to a minimum, using standard values for the unsupervised graph algorithms. For wordgraph algorithms, Textrank and Topical PageRank, length of the sliding window was set to

[10]Due to copyright issues, the data cannot be published publicly: researchers only can obtain the data (including annotations and candidate keyphrases) after contacting the authors and signing a non-disclosure agreement.

10 tokens and a damping factor of 0.85 for TextRank and 0.7 for Topical PageRank was chosen.

Development of classifiers was done by crossfolding the trainingdata four times. Hyperparameters for boosted trees are the number of trees and their depth, for the SVM classifier we tune the regularization. We optimized for micro-averaged F_1 scores on the held-out folds.

Note that some of the keyphrases assigned by the annotators are not extracted by the candidate generator. These are filtered out from the train set (as indicated by the processing step in the top right of Fig. 4.7), to prevent classifiers from overfitting on forms of keyphrases that are not generated by the candidate generator. Such keyphrases that do not match the candidate generation pattern are however included in the ground truth set of keywords for the test collection, so for most documents a recall of 100% is not attainable.

4.4.2 Evaluation Setup using Multiple Opinions

Several options arise when evaluating keyphrases for documents with multiple opinions and depending on the goal or application of the keyphrases, different requirements are to be met or preferred by the keyphrase extractor's output. We propose two quite different evaluation scenarios, as well as a short motivation for both of them, followed by experimental results. As we will demonstrate, deciding on one of these scenarios strongly influences the scores and may lead to differently ranked keyphrase extractors.

A straightforward way is to create a reference set of keyphrases by pooling the annotated keyphrases by the different judges. The main advantage of this pooling approach is the increased robustness when measuring precision. Different annotators can select different representatives for identical concepts in an article. This way, the precision of the keyphrase extractor remains when making a specific choice of keyphrases, provided it covers the central concepts present in the reference set. However, this scenario suffers from two drawbacks. First, it does not penalize a possible lack of diversity among the predicted keyphrases, and second, it is hard to interpret the resulting metrics based on aggregated keyphrases from the point of view of a single annotator. This scenario is preferable when keyphrases are applied for visual purposes, e.g., to provide an overview of content by highlighting all keyphrases, or to get an estimate of the overall precision of the extractor regardless of redundancy in the output of the extractors. The second scenario, discussed next, avoids these drawbacks.

In the second scenario, each set of keyphrases from a specific annotator is treated as an independent target set. In this case, averaging over the obtained evaluation metrics corresponds to measuring the expected performance for a random annotator, if we can assume that the annotator population is represented by the set of annotators. This evaluation scenario comes closer to the purpose of summarization by keyphrases, which

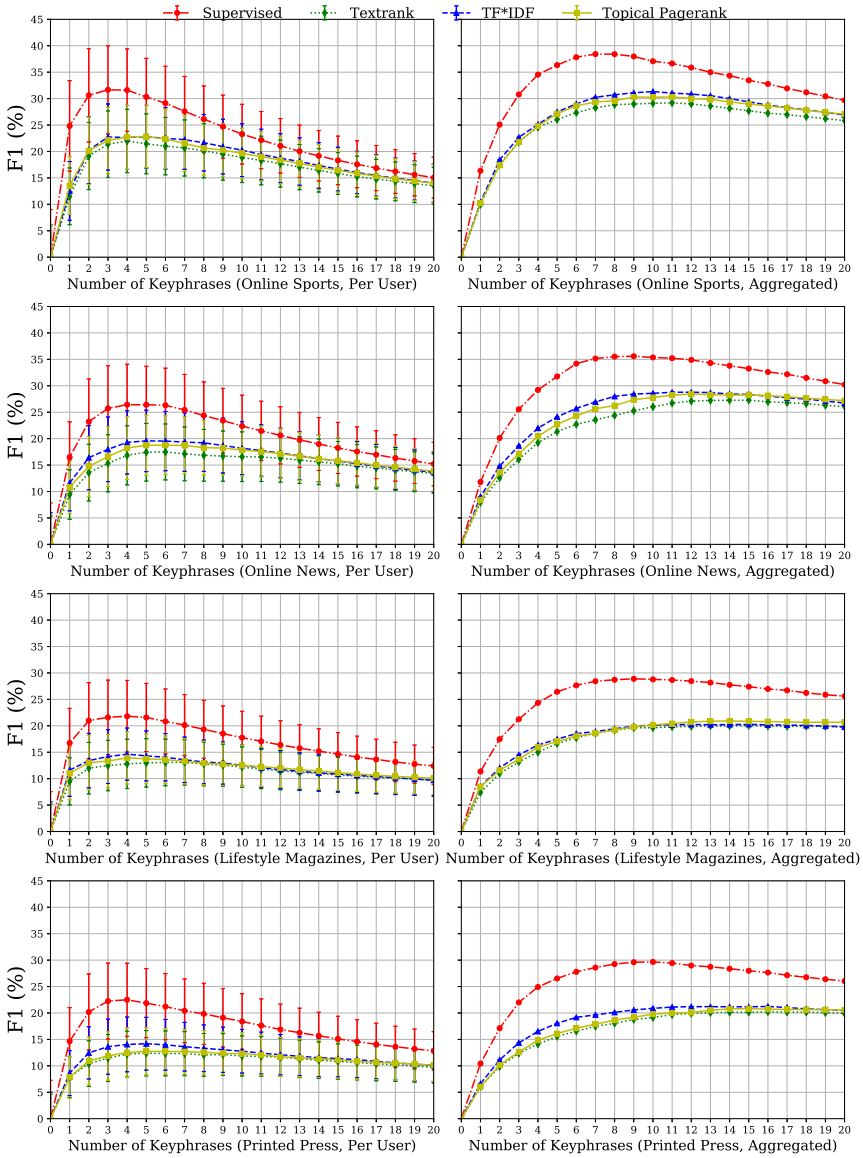


Figure 4.8: Plots on the left show micro-averaged F_1 scores (with error bars showing standard deviation) for different fixed amounts of assigned keyphrases for different annotators. The right column shows the same models evaluated on aggregated collections of keyphrases.

is a more common goal of keyphrases and and in line with the instructions given to the annotators. In this setting, extractors are rewarded for the extraction of small but diverse sets of keyphrases as annotated by different annotators. We perform this annotator based evaluation by averaging the scores over different annotators per document and measuring the standard deviation on this average.

To match predicted keyphrases with the reference keyphrases, we follow the traditional evaluation scheme for keyphrase extraction [12] by exactly matching keyphrases from the golden answer set with those provided by the automatic extractors without stemming, and apply a standard rank- or set-based metric. We measure the micro-averaged precision for the 5 (precision@5) top ranked or most confident keyphrases. Note that 5 is approximately the average amount of keyphrases assigned by a single annotator to a document and is the number of keyphrases the content providers agreed to assign to each document. To evaluate from a set-based perspective, we measure the micro-averaged F_1 from precision and recall per document after setting a threshold (the same for all documents) on the confidence values predicted by the extractors that optimizes F_1 scores on the development set .

In Tables 4.5 and 4.6, we show results for these two different approaches to evaluation, i.e., the first scenario with aggregated target collections (*Aggr.*), and the second scenario with scores averaged per annotator (*Av.*$\pm$*Stdv.*) for the precision at 5 extracted keyphrases per document (precision@5) in Table 4.5 and F_1 scores at a tuned threshold in Table 4.6. We define precision, recall and F_1 as follows:

$$\text{precision} = \frac{|\text{annotated keyphrases} \cap \text{extracted keyphrases}|}{|\text{extracted keyphrases}|} \tag{4.10}$$

$$\text{recall} = \frac{|\text{annotated keyphrases} \cap \text{extracted keyphrases}|}{|\text{annotated keyphrases}|} \tag{4.11}$$

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.12}$$

In Fig. 4.8, we plot F_1 as a function of a fixed amount of selected keyphrases per document to further illustrate the difference between these two ways of evaluating. Graphs in the left column show results for averaging over sets of keyphrases by the different annotators with standard deviations, graphs on the right show evaluations on the aggregated set of keyphrases.

4.4.3 Comparison of different techniques

A first observation is that, in terms of performance of different AKE approaches, *supervised* models using baseline features (see Section 4.3.3.2)

Table 4.5: Experimental results over different unsupervised and supervised models. The **precision at 5** selected keyphrases is evaluated on an aggregated set of keyphrases from different annotator (Aggr.) and for scores averaged over different annotators with standard deviation (Av.±Stdv.). Development scores for supervised classifiers are included between brackets.

Test Collection→	**Online Sports**		**Online News**	
Model ↓	Aggr.	Av.±Stdev.	Aggr.	Av.±Stdev.
TextRank	39.6	18.8 ±5.1	37.5	16.3 ±5.2
TF*IDF	41.9	20.0 ±5.4	42.8	18.4 ±5.7
Topical PageRank	41.4	20.0 ±5.3	40.0	17.6 ±5.5
XGBoost	(57.9) 55.2	(27.5) 26.7 ±6.9	(56.6) 56.0	(24.3) 24.6 ±7.2
XGBoost + $f_{\text{TF*IDF}}$	(59.2) 56.2	(28.0) **27.0** ±**6.9**	(57.4) 56.0	(24.6) 24.9 ±7.2
XGBoost+ $f_{\text{TF*IDF}}$, f_{LDA}	(60.5) 55.8	(28.2) 26.9 ±6.9	(59.3) **56.4**	(25.3) 24.9 ±7.3
SVM + $f_{\text{TF*IDF}}$, f_{LDA}	(53.6) 51.8	(25.6) 24.7 ±6.5	(52.2) 52.4	(22.9) 23.1 ±6.8
Wikipedia Background:				
TF*IDF	40.3	19.6 ±5.4	40.1	17.2 ±5.6
Topical PageRank	40.8	19.6 ±5.4	38.1	16.7 ±5.3
XGBoost + $f_{\text{TF*IDF}}$, f_{LDA}	(60.0) **56.2**	(28.3) 26.9 ±6.9	(59.8) **56.4**	(25.4) **25.0** ±**7.2**
Test Collection→	**Lifestyle Magazines**		**Printed Press**	
Model ↓	Aggr.	Av.±Stdev.	Aggr.	Av.±Stdev.
TextRank	28.8	11.8 ±4.2	28.0	11.5 ±4.1
TF*IDF	30.3	13.1 ±4.4	32.7	13.3 ±4.8
Topical PageRank	29.2	12.5 ±4.3	28.6	11.7 ±4.2
XGBoost	(47.1) 45.6	(20.6) 19.4 ±6.2	(48.3) 45.8	(20.8) 19.4 ±6.3
XGBoost + $f_{\text{TF*IDF}}$	(47.2) 45.1	(20.8) 19.3 ±6.0	(48.2) 46.8	(20.5) 19.8 ±6.3
XGBoost+ $f_{\text{TF*IDF}}$, f_{LDA}	(48.1) 46.0	(20.9) 19.7 ±6.1	(49.8) **47.5**	(21.1) **20.2** ±**6.3**
SVM + $f_{\text{TF*IDF}}$, f_{LDA}	(43.3) 42.0	(19.3) 18.0 ±5.7	(42.6) 41.0	(18.0) 17.2 ±5.7
Wikipedia Background:				
TF*IDF	26.9	11.2 ±4.0	29.6	12.1 ±4.4
Topical PageRank	27.5	11.6 ±4.1	28.4	11.4 ±4.2
XGBoost + $f_{\text{TF*IDF}}$, f_{LDA}	(48.8) **46.4**	(21.3) **20.0** ±**6.3**	(49.4) 46.8	(21.2) 19.8 ±6.4

outperform each of the standard unsupervised techniques on every test collection by a margin for the different metrics. These results highlight the (perhaps unsurprising) need for supervision and feature design. Models show improvement using contextual information ($f_{\text{TF*IDF}}$, f_{LDA}). Also for unsupervised models, techniques like TF*IDF and Topical PageRank which include background information generally perform better than those that do not, like TextRank. For statistical classifiers, the gradient boosted decision tree outperforms the linear classifier on each occasion.

For the TF*IDF and Topical PageRank models, using IDFs or topic models outperform those inferred from the more general Wikipedia background collection. This is less the case when they are used as features in the supervised models.

4.4.4 Comparison of different test collections

Between test collections there is a clear distinction in performance by keyphrase extractors for the different types of content. Precision@5 and F_1 scores for keyphrases predicted on *Online Sports* content can be up to 10%

Table 4.6: Experimental results for different unsupervised and supervised models. The macro-averaged $\mathbf{F_1}$ **measure** selected keyphrases is evaluated on an aggregated set of keyphrases from different annotators (Aggr.) and for scores averaged for different annotators with standard deviation (Av.±Stdv.). Development scores for supervised classifiers are included between brackets.

Test Collection→	**Online Sports**		**Online News**	
Model ↓	Aggr.	Av.±Stdev.	Aggr.	Av.±Stdev.
TextRank	26.0	22.1 ±5.9	21.3	17.2 ±5.1
TF*IDF	27.5	22.6 ±5.9	24.1	19.3 ±5.9
Topical PageRank	27.1	22.7 ±5.8	22.8	18.6 ±5.5
XGBoost	(35.9) 36.0	(31.5) 32.4 ±8.3	(29.2) 31.6	(26.0) 26.7 ±7.4
XGBoost + $f_{\text{TF*IDF}}$	(36.7) 36.6	(32.1) 32.1 ±8.1	(29.7) 31.7	(26.3) 26.8 ±7.5
XGBoost+ $f_{\text{TF*IDF}}$, f_{LDA}	(37.5) 36.4	(32.3) 32.6 ±8.2	(30.8) **31.8**	(27.1) **27.1** ±7.7
SVM + $f_{\text{TF*IDF}}$, f_{LDA}	(33.2) 33.9	(29.4) 27.3 ±7.4	(27.1) 29.5	(24.5) 23.8 ±7.2
Wikipedia Background:				
TF*IDF	26.4	22.7 ±6.1	22.6	18.1 ±5.7
Topical PageRank	26.9	22.6 ±6.0	21.7	17.8 ±5.1
XGBoost + $f_{\text{TF*IDF}}$, f_{LDA}	(37.2) **36.7**	(32.4) **33.1** ±**8.6**	(30.9) **31.8**	(27.2) **27.1** ±**7.6**
Test Collection→	**Lifestyle Magazines**		**Printed Press**	
Model ↓	Aggr.	Av.±Stdev.	Aggr.	Av.±Stdev.
TextRank	16.6	13.6 ±5.0	15.6	12.8 ±4.3
TF*IDF	17.5	15.4 ±5.1	18.1	14.2 ±4.8
Topical PageRank	17.0	14.8 ±5.1	16.1	12.9 ±4.3
XGBoost	(26.2) 26.5	(23.0) 23.0 ±7.4	(25.8) 25.6	(22.8) 21.3 ±7.0
XGBoost + $f_{\text{TF*IDF}}$	(26.2) 26.0	(23.2) 23.1 ±7.4	(25.8) 26.1	(22.5) 22.0 ±7.0
XGBoost+ $f_{\text{TF*IDF}}$, f_{LDA}	(26.7) 26.4	(23.3) 23.1 ±7.2	(26.6) **26.5**	(23.2) **22.7** ±**7.1**
SVM + $f_{\text{TF*IDF}}$, f_{LDA}	(24.1) 24.2	(21.6) 19.5 ±6.7	(22.7) 22.4	(19.7) 17.9 ±6.1
Wikipedia Background:				
TF*IDF	15.5	13.0 ±4.7	16.4	13.2 ±4.7
Topical PageRank	16.2	13.8 ±4.7	15.8	12.6 ±4.3
XGBoost + $f_{\text{TF*IDF}}$, f_{LDA}	(27.1) **26.8**	(23.7) **23.3** ±**7.2**	(26.4) 26.1	(23.2) 22.3 ±7.0

higher than those for *Lifestyle* and *Printed Press*. An explanation for this may be the focus on entities in these Sports documents, and higher annotator agreement for these documents. These types of keyphrases are covered well by candidate generators and are modeled well by the features. Scores for *Online News* are overall better than *Lifestyle* and *Printed Press*.

The optimal number of keyphrases for aggregated target sets is eight, and for per-annotator sets is four. This difference seems reasonable, given that a single annotator on average assigns about 5 keyphrases, whereas the per-annotator sets are aggregates of 5 annotations.

As a result from the high levels of disagreement between annotators, standard deviations on averaged scores are equally high, even more so for supervised models than unsupervised models. Optimal cut-off confidence thresholds for optimal F_1 are highly dependent on the evaluation setting (aggregated versus annotators based) as is apparent from Fig. 4.8. While scores calculated for aggregated sets of keyphrases (*Aggr.*) are not be compared with those averaged over the different annotators (*Av.*), the difference between them is most notable for the precision@5 metric. Precision@5 scores are generally much higher for aggregated sets as these contain much

more keyphrases as they include the same semantic concepts in different forms. On average precision@5 for an automated extractor is around 50%, whereas this value drops to around 25% when evaluated for annotators separately. For the common purpose of summarization by keyphrases, we advocate the use of multiple opinions per document for evaluation. As the task is inherently objective, obtaining scores with low deviations is a desirable aspect of a keyphrase extractor, as this means the keyphrase sets satisfy different opinions better. When keyphrases are used for visual purposes, a better objective is to optimize the score for aggregated sets of keyphrases.

4.4.5 Training set size

In Section 4.4.2 the need for supervision in automatic keyphrase extraction was highlighted. In this section we study the annotation effort versus performance. Figure 4.9 plots the performance of supervised models with contextual features (XGBoost $+f_{\text{TF*IDF}} + f_{\text{LDA}}$) for different amounts of annotated documents in the training data. We use limited sets of training data (10, 25, 50, 100 and 300 documents) and measure the annotator-averaged F_1 score using the models. This demonstrates the rapid increase in supervised performance over the unsupervised models. From a minimum of 25 annotated documents, the F_1 measure exceeds each of the unsupervised models. Another observation is that the optimal performance is reached quite rapidly: a maximum value is obtained for as few as a hundred annotated documents. On the one hand, this shows that the annotation cost for a supervised system that significantly outperforms the best unsupervised systems is quite low. On the other hand, it highlights the need for more descriptive features to further improve supervised keyphrase extraction.

4.4.6 Training data from multiple opinions

Previous sections focused on the effect of multiple opinions on the *evaluation* of keyphrase extraction. Figure 1.1 shows the resulting F_1score for the different subcollections, comparing a supervised method (XGBoost + $f_{\text{TF*IDF}}+ f_{\text{LDA}}$) with an unsupervised method (Topical PageRank), as a function of the number of training items (ranging from 5 to 300 documents). For the supervised method, we also show the influence of multiple opinions during *training* ('5 Annotators' vs. '1 Annotator) The '5 Annotators' case makes use of the aggregated set of annotated keyphrases from all 5 annotators for each document. We observe a limited increase in F_1 performance by aggregating keyphrase sets. This is partly due to the increasing amount of positive training data. An additional explanation is the following: for training collections generated by single readers, or document authors, it is likely that many candidates not tagged as keyphrase, might be seen as keyphrases by others. Yet during training they are implicitly considered

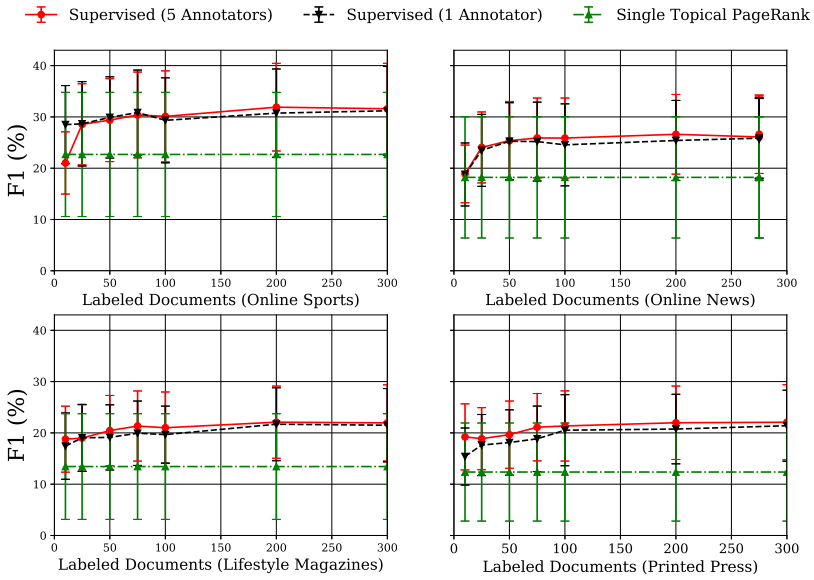


Figure 4.9: Supervised model (XGBoost + $f_{\text{TF*IDF}}$+ f_{LDA}) versus various unsupervised model (Topical PageRank) for different amounts of training data.

as negative cases. When multiple opinions are not available, an elegant solution is to recast the problem into a positive versus unlabeled learning setting [38, 39].

4.4.7 Effect of Document Length

Finally we study the influence of document length on extraction performance. Figure 4.10 shows F_1 averaged per annotator for the supervised models as a function of document length. The overall trend is that scores get lower for longer documents, most notably for *Lifestyle* content. Intuitively, this is not surprising, since a longer document will produce more candidate phrases and it thus becomes more difficult to pick the (about 5) correct ones.

4.5 Guidelines for Automatic Keyphrase Extraction

In this paper, we presented a number of large, new collections for evaluation of Automatic Keyphrase Extraction, with multiple opinions per docu-

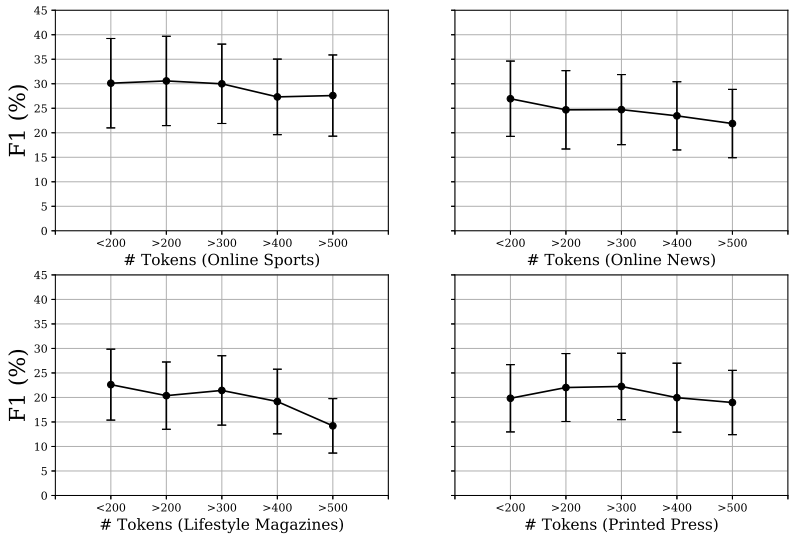


Figure 4.10: Annotator-averaged F_1 for the supervised model (XGBoost + $f_{\text{TF*IDF}}$+ f_{LDA}) versus document length.

ment. A panel of more than 350 annotators provided sets of keyphrases for different types of content from *online news*, *online sports*, *lifestyle magazines* and *printed press*. We were able to quantify the subjectivity of the keyphrase extraction task in terms of annotator disagreement, thanks to availability of multiple opinions per annotated document. As shown, this has important consequences on evaluation and training of keyphrase extractors. We studied different ways of assessing keyphrase extractor output for a number of existing supervised and unsupervised techniques. Our evaluation experiments demonstrated the importance of a suitable candidate generation strategy, and the superior effectiveness of supervised models over unsupervised models, even for low annotation efforts. When training on documents with multiple opinions, a small increase in performance is found using aggregated sets of keyphrases for training.

Many challenges and opportunities for effective keyphrase extraction remain. To conclude, we present several guidelines one should take into account when automatically extracting keyphrases and future research opportunities.

- **Candidate generation:** Proper candidate generation cannot be underestimated. Depending on the type of the documents, candidate phrases need to cover the keyphrases assigned by annotators while

keeping the ratio of candidates versus keyphrases as low as possible. Figure 4.6 indicates that many valid keyphrases can be lost in this stage by over-filtering or missing crucial keyphrase forms while generating too many candidates which seldom appear as a keyphrase.

- **Feature Design:** A crucial aspect of keyphrase extraction remains supervision and feature design. As Fig. 4.9 shows, performance of supervised classifiers tends to stagnate for a relatively low amount of training data, which indicates limited expressiveness by the features. A possibility for future research is the use of neural network classifiers for more sophisticated representations of keyphrases.

- **Evaluation:** Our evaluation show the frailty of current keyphrase evaluation and potentially large fluctuations in scores depending on what sets of annotations are used for evaluation. As [40] already noted, a more suitable evaluation for keyphrase extractors would be to let annotators compare sets of keyphrases output by different models. A downside of this setting is that different models need to be evaluated separately, and fine-tuning towards individual opinions is impractical.

- **Task subjectivity:** Low agreement between annotators on keyphrases as demonstrated in Table 4.3 shows that keyphrase extraction remains a highly subjective natural language processing task with large consequences for evaluation and training.

- **Reranking:** A topic that received less attention in this first evaluation is topic coverage in keyphrase sets. Some keyphrase extraction systems have been proposed with ways to optimize the coverage of topics in sets of keyphrases [41].

The aim of this work was to underline the importance of these issues, and therefore we make our new test collections available for academic use, to encourage research on better evaluation and extraction techniques of keyphrases, addressing a number of open issues in this area.

4.A Supervised Keyphrase Extraction as Positive Unlabeled Learning

In follow-up work, making use of the datasets presented above, we study the influence of multiple opinions of keyphrases on training and evaluation. Multiple opinions are expensive to obtain but are necessary for reliable evaluation and training. We show that, by rephrasing the problem of kephrase extraction as positive-unlabeled learning we obtain scores which approximate scores from the ideal case of classifiers trained on multiple opinions, even when only based on single annotations, by applying a reweighting strategy of unlabeled candidates and strategies to counter the imbalance or noise of the training collections.

⋆ ⋆ ⋆

L. Sterckx, C. Caragea, T. Demeester and C. Develder

Presented at the Conference on Empirical Methods in Natural Language Processing, Austin (Texas), USA, 2016.

Abstract In previous chapters, the problem of noisy and unbalanced training data for supervised keyphrase extraction results from the subjectivity of keyphrase assignment, which we quantify by crowdsourcing keyphrases for articles with many annotators per document. We show that annotators exhibit substantial disagreement, meaning that single annotator data could lead to very different training sets for supervised keyphrase extractors. Thus, annotations from single authors or readers lead to noisy training data and poor extraction performance of the resulting supervised extractor. We provide a simple but effective solution to still work with such data by reweighting the importance of unlabeled candidate phrases in a two stage Positive Unlabeled Learning setting. We show that performance of trained keyphrase extractors approximates a classifier trained on articles labeled by multiple annotators, leading to higher average F_1scores and better rankings of keyphrases. We apply this strategy to a variety of test collections from different backgrounds and show improvements over strong baseline models.

4.A.1 Introduction

Keyphrase extraction is the task of extracting a selection of phrases from a text document to concisely summarize its contents. Applications of keyphrases range from summarization [2] to contextual advertisement [4] or simply as aid for navigation through large text corpora.

Existing work on automatic keyphrase extraction can be divided in supervised and unsupervised approaches. While unsupervised approaches

are domain independent and do not require labeled training data, supervised keyphrase extraction allows for more expressive feature design and is reported to outperform unsupervised methods on many occasions [42, 43]. A requirement for supervised keyphrase extractors is the availability of labeled training data. In literature, training collections for supervised keyphrase extraction are generated in different settings. In these collections, keyphrases for text documents are either supplied by the authors or their readers. In the first case, authors of academic papers or news articles assign keyphrases to their content to enable fast indexing or to allow for the discovery of their work in electronic libraries [8, 16, 44]. Other collections are created by crowdsourcing [45] or based on explicit deliberation by a small group of readers [11]. A minority of test collections provide multiple opinions per document, but even then the amount of opinions per document is kept minimal [46].

The traditional procedure for supervised keyphrase extraction is reformulating the task as a binary classification of keyphrase candidates. Supervision for keyphrase extraction faces several shortcomings. Candidate phrases (generated in a separate candidate generation procedure), which are not annotated as keyphrases, are seen as non-keyphrase and are used as negative training data for the supervised classifiers. First, on many occasions these negative phrases outnumber true keyphrases many times, creating an unbalanced training set [16, 42]. Second, as Frank et al. [16] noted: authors do not always choose keyphrases that best describe the content of their paper, but they may choose phrases to slant their work a certain way, or to maximize its chance of being noticed by searchers. Another problem is that keyphrases are inherently subjective, i.e., keyphrases assigned by one annotator are not the only correct ones [46]. These assumptions have consequences for training, developing and evaluating supervised models. Unfortunately, a large collection of annotated documents by reliable annotators with high overlap per document is missing, making it difficult to study disagreement between annotators or the resulting influence on trained extractors, as well as to provide a reliable evaluation setting. In this paper, we address these problems by creating a large test collection of articles with many different opinions per article, evaluate the effect on extraction performance, and present a procedure for supervised keyphrase extraction with noisy labels.

4.A.2 Noisy Training Data for Supervised Keyphrase Extraction

A collection of online news articles and lifestyle magazine articles was presented to a panel of 357 annotators of various ages and backgrounds, (selected and managed by iMinds - Living Labs[11]) who were trained to select

[11] https://www.iminds.be/en/succeed-with-digital-research/proeftuinonderzoek/

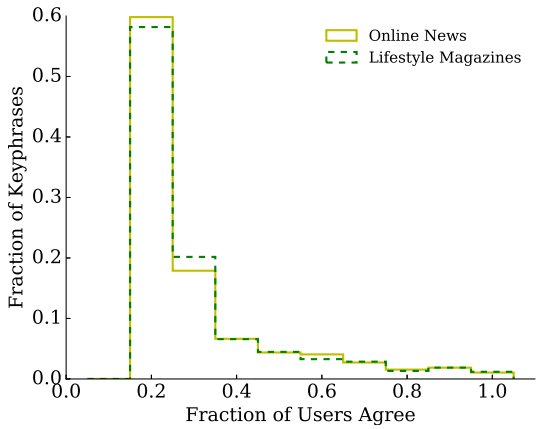


Figure 4.11: This plot shows the fraction of all keyphrases from the training set agreed upon versus the fraction of all annotators.

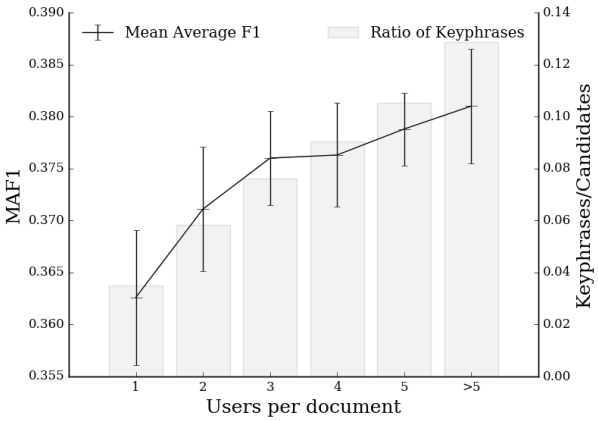


Figure 4.12: Effect of overlap on extraction performance.

a *limited number* of *short* phrases that concisely reflect the documents' contents. No prior limits or constraints were set on the amount, length, or form of the keyphrases. Each document was presented multiple times to different users. Each user was assigned with 140 articles, but was not required to finish the full assignment. The constructed training collections have on average six and up to ten different opinions per article.

We visualize the agreement on single keyphrases in Figure 4.11, which shows the fraction of annotated keyphrases versus agreement by the complete set of readers. Agreement on keyphrases appears low, as a large frac-

tion of all keyphrases assigned to documents (>50%) are only assigned by single annotators. We note that different sets of keyphrases by different annotators are the result of the subjectiveness of the task, of different interpretations by the annotators of the document, but also because of semantically equivalent keyphrases being annotated in different forms, e.g., “Louis Michel" vs. “Prime Minister Louis Michel" or “Traffic Collision" vs. “Car Accident".

The observation in Figure 4.11 has important consequences for training models on keyphrases annotated by a single annotator, since other annotators may have chosen some among the ones that the single selected annotator did not indicate (and hence these should not be used as negative training data).

A single annotator assigning keyphrases to 100 documents results on average in a training set with 369 positive training instances and 4,981 negative training instances generated by the candidate extractor. When assigning these 100 documents to 9 other annotators, the amount of positive instances increases to 1,258 keyphrases, which means that labels for 889 keyphrase candidates, or 17% of the original negative candidates when training on annotations by a single annotator, can be considered noise and relabeled. As a result, ratios of positive to negative data also change drastically. We visualize the effect of using training data from multiple annotators per document in Figure 4.12. Classifiers trained on the aggregated training collection with multiple opinions (using all assigned keyphrases at least once as positive training data) perform better on held-out test collections containing only keyphrases of high agreement (assigned by > 2 annotators).

When using keyphrases from many different annotators per document, the amount of positive candidates increases and as a result, the Macro Average F_1 (MAF_1) of the corresponding classifier. We detail our experimental setup and supervised classifier in Section 4.A.3.1.

4.A.3 Reweighting Keyphrase Candidates

Observations described in Section 4.A.2 indicate that unlabeled keyphrase candidates are not reliable as negative examples by default. A more suitable assumption is to treat supervised keyphrase extraction as Positive Unlabeled Learning, i.e., an incomplete set of positive examples is available as well as a set of unlabeled examples, of which some are positive and others negative. This topic has received much attention as it has many applications [47, 48], but has not been explored for supervised keyphrase extraction. We base our approach on work by Elkan and Noto [38] and modify the supervised extractor by assigning individual weights to training examples. Instead of assuming the noise to be random, we assign weights depending on the document and the candidate.

Table 4.7: String relation features for coreference resolution

Feature	Definition
Head match	$head_{keyphrase} == head_{candidate}$
Extent match	$extent_{keyphrase} == extent_{candidate}$
Substring	$head_{keyphrase}$ substring of $head_{candidate}$
Alias	$acronym(head_{keyphrase}) == head_{candidate}$

By reweighting importance of training samples, we seek to mimic the case of multiple annotators, to model the uncertainty of negative keyphrase candidates, based only on annotations by a single annotator. In a first stage, we train a classifier on the single annotator data and use predictions on the negative or unlabeled candidates, to reweigh training instances. The reweighted training collection is then used to train a second classifier to predict a final ranking or the binary labels of the keyphrase candidates.

Positive examples are given unit weight and unlabeled examples are duplicated; one copy of each unlabeled keyphrase candidate x is made positive with weight $w(x) = P(keyphrase|x, s = 0)$ and the other copy is made negative with weight $1 - w(x)$ with s indicating whether x is labeled or not.

Instead of assigning this weight as a constant factor of the predictions by the initial classifier as in Elkan and Noto [38], we found that two modifications allow improving the weight estimate, $w(x) \leq 1$. We normalize probabilities $P(keyphrase, x, s = 0)$ to candidates not included in the initial set of keyphrases per document. Besides this self-predicted probability, we include a simple measure indicating pairwise coreference between unlabeled candidates and known keyphrases in a function Coref(candidate,keyphrase)$\in \{0, 1\}$,

returning 1 if one of the binary indicator features, presented in [49] and shown in Table 4.7, is present. In this description, the term *head* means the head noun phrase of a candidate or keyphrase and the *extent* is the largest noun phrase headed by the head noun phrase. The self-predicted probability is summed with the output of the coreference resolver and the final weight becomes:

$$w(x) = min\left(1, \frac{P(keyphrase|x)}{\max_{(x',s=0)\in d} P(keyphrase|x')} + \max_{\forall keyphrase \in d} \text{Coref}(x, keyphrase)\right)$$

with d being a document from the training collection.

4.A.3.1 Experiments and Results

Hasan and Ng [50] have shown that techniques for keyphrase extraction are inconsistent and need to be tested across different test collections. Next

to our collections with multiple opinions (*Online News* and *Lifestyle Magazines*), we apply the reweighting strategy on test collections with sets of author-assigned keyphrases: two sets from CiteSeer abstracts from the World Wide Web Conference (*WWW*) and Knowledge Discovery and Data Mining (*KDD*), similar to the ones used in [8]. The *Inspec* dataset is a collection of 2,000 abstracts commonly used in keyphrase extraction literature, where we use the ground truth phrases from controlled vocabulary [44]. Descriptive statistics of these test collections are given in Table 4.8.

We use a rich feature set consisting of statistical, structural, and semantic properties for each candidate phrase, that have been reported as effective in previous studies on supervised extractors [7, 16, 44]: (i) term frequency, (ii) number of tokens in the phrase, (iii) length of the longest term in the phrase, (iv) number of capital letters in the phrase, (v) the phrase's POS-tags, (vi) relative position of first occurrence, (vii) span (relative last occurrence minus relative first occurrence), (viii) TF*IDF (IDF's trained on large background collections from the same source) and (ix) Topical Word Importance, a feature measuring the similarity between the word-topic topic-document distributions presented in [22], with topic models trained on background collections from a corresponding source of content.

As classifier we use gradient boosted decision trees implemented in the XGBoost package [35]. During development, this classifier consistently outperformed Naive Bayes and linear classifiers like logistic regression or support vector machines.

We compare the reweighting strategy with uniform reweighting and strategies to counter the imbalance or noise of the training collections, such as subsampling, weighting unlabeled training data as in [38], and self-training in which only confident initial predictions are used as positive and negative data. For every method, global thresholds are chosen to optimize the macro averaged F_1 per document (MAF_1). Next to the threshold sensitive F_1, we report on ranking quality using the Precision@5 metric.

Results are shown in Table A.6 with five-fold cross-validation. To study the effect of reweighting, we limit training collections during folds to 100 documents for each test collection. Our approach consistently improves on single annotator trained classifiers, on one occasion even outperforming a training collection with multiple opinions. Compensating for imbalance and noise tends to have less effect when the ratio of keyphrases versus candidates is high (as for *Inspec*) or training collection is very large. When the amount of training documents increases, the ratio of noisy versus true negative labels drops. As future work we suggest using a separate coreference resolver trained on a corpus annotated with coreferential relations and a coreference resolution system for Dutch [51].

4.A.4 Conclusion

It has been suggested that keyphrase annotation is highly subjective. We present two data sets where we purposely gathered multiple annotations of the same document, as to quantify the limited overlap between keyphrases selected by different annotators. We suggest to treat non-selected phrases as *unlabeled* rather than *negative* training data. We further show that using multiple annotations leads to more robust automatic keyphrase extractors, and propose reweighting of single annotator labels based on probabilities from a first-stage classifier. This reweighting approach outperforms other single-annotator state-of-the-art automatic keyphrase extractors on different test collections, when we normalize probabilities per document and include co-reference indicators.

	Online News	**Lifestyle Magazines**	**WWW**	**KDD**	**Inspec**
Type	Sports Articles	Fashion, Lifestyle	WWW Paper Abstracts	KDD Paper Abstracts	Paper Abstracts
# Documents	1,259	2,202	1,895	1,011	500
# Keyphrases	19,340	29,970	3,922	1,966	4,913
⌀ Keyphrases/User	5.7	4.7	/	/	/
⌀ Keyphrases/Document	15.4	13.7	2.0	1.8	9.8
⌀ Tokens/Document	332	284	164	195	134
⌀ Candidate Keyphrases/Doc.	52	49	47	54	34
1/2/3/3+ -gram distribution (%)	55/27/9/9	58/25/9/8	63/27/8/2	60/28/9/3	13/53/25/9

Table 4.8: Description of test collections.

	Online News		**Lifestyle Magazines**		**WWW**		**KDD**		**Inspec**	
	MAF_1	P@5	MAF_1	P@5	MAF_1	P@5	MAF_1	P@5	MAF_1	P@5
Single Annotator	.364	.416	.294	.315	.230	.189	.266	.200	.397	.432
Multiple Annotators	.381	.426	.303	.327	/	/	/	/	/	/
Self Training	.366	.417	.301	.317	.236	.190	.269	.196	.401	**.434**
Reweighting [38]	.364	.417	.297	.313	.238	.189	**.275**	**.201**	.401	.429
Reweighting +Norm +Coref	**.374**	**.419**	**.305**	**.322**	**.245**	**.194**	**.275**	.200	**.402**	**.434**

Table 4.9: Mean average F_1score per document and precision for five most confident keyphrases on different test collections.

References

[1] P. D. Turney. *Learning algorithms for keyphrase extraction*. Information retrieval, 2(4):303–336, 2000.

[2] E. D'Avanzo, B. Magnini, and A. Vallin. *Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004*. In Proceedings of the 2004 DUC, 2004.

[3] K. M. Hammouda, D. N. Matute, and M. S. Kamel. *Corephrase: Keyphrase extraction for document clustering*. In Machine Learning and Data Mining in Pattern Recognition, pages 265–274. Springer, 2005.

[4] W.-t. Yih, J. Goodman, and V. R. Carvalho. *Finding advertising keywords on web pages*. In Proceedings of the 15th international conference on World Wide Web, pages 213–222. ACM, 2006.

[5] A. Hulth. *Improved automatic keyword extraction given more linguistic knowledge*. Proceedings of the 2003 conference on Empirical Natural language Processing, (2000), 2003. Available from: http://dl.acm.org/citation.cfm?id=1119383.

[6] P. Lopez and L. Romary. *HUMB: Automatic key term extraction from scientific articles in GROBID*. In Proceedings of the 5th international workshop on semantic evaluation, pages 248–251. Association for Computational Linguistics, 2010.

[7] S. N. Kim and M.-Y. Kan. *Re-examining automatic keyphrase extraction approaches in scientific articles*. In Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications, pages 9–16. Association for Computational Linguistics, 2009.

[8] F. A. Bulgarov and C. Caragea. *A Comparison of Supervised Keyphrase Extraction Models*. In Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pages 13–14, 2015. Available from: http://doi.acm.org/10.1145/2740908.2742776, doi:10.1145/2740908.2742776.

[9] C. Wartena, R. Brussee, and W. Slakhorst. *Keyword Extraction Using Word Co-occurrence*. 2010 Workshops on Database and Expert Systems Applications, pages 54–58, August 2010. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5592000, doi:10.1109/DEXA.2010.32.

[10] Z. Liu, W. Huang, Y. Zheng, and M. Sun. *Automatic keyphrase extraction via topic decomposition*. In Proceedings of the 2010 Conference on EMNLP, pages 366–376, 2010.

[11] X. Wan and J. Xiao. *Single Document Keyphrase Extraction Using Neighborhood Knowledge*. In Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI 2008, pages 855–860, 2008. Available from: http://dl.acm.org/citation.cfm?id=1620163.1620205.

[12] K. S. Hasan and V. Ng. *Automatic keyphrase extraction: A survey of the state of the art*. Proceedings of the Association for Computational Linguistics (ACL), Baltimore, Maryland: Association for Computational Linguistics, 2014.

[13] J. Bowman. *Essential Cataloguing*. Facet Pub., 2003. Available from: https://books.google.be/books?id=C-7gAAAAMAAJ.

[14] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum. *SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications*. ArXiv e-prints, April 2017. arXiv:1704.02853.

[15] W. X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.-P. Lim, and X. Li. *Topical Keyphrase Extraction from Twitter*. In Proceedings of the 49th Annual Meeting of the ACL: HLT- Volume 1, HLT '11, pages 379–388, Stroudsburg, PA, USA, 2011. Available from: http://dl.acm.org/citation.cfm?id=2002472.2002521.

[16] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-manning. *Domain Specific Keyphrase Extraction*. In Proceedings of the 16th International Joint Conference on AI, pages 668–673, 1999.

[17] J. L. Fleiss. *Measuring nominal scale agreement among many raters.* Psychological bulletin, 76(5):378, 1971.

[18] B. Lievens, B. Baccarne, C. Veeckman, S. Logghe, and D. Schuurman. *Drivers For End-users' Collaboration In Participatory Innovation Development And Living Lab Processes*. In 17th ACM Conference on Computer Supported Cooperative Work and Social Computing, 2014.

[19] O. Medelyan and I. Witten. *Thesaurus based automatic keyphrase indexing*. In Proceedings of the 6th ACM/IEED-CS joint conference on Digital libraries, pages 296–297, 2002.

[20] M. Grineva, M. Grinev, and D. Lizorkin. *Extracting key terms from noisy and multitheme documents*. WWW 2009 MADRID! Track: Semantic/Data Web / Session: Mining for Semantics, pages 661–670, 2009. Available from: http://dl.acm.org/citation.cfm?id=1526798.

[21] R. Mihalcea and A. Csomai. *Wikify!: linking documents to encyclopedic knowledge*. CIKMâĂŹ07, November 6âĂŞ8, 2007, Lisboa, Portugal, (July), 2007. Available from: http://dl.acm.org/citation.cfm?id=1321475.

[22] L. Sterckx, T. Demeester, J. Deleu, and C. Develder. *Topical Word Importance for Fast Keyphrase Extraction*. In Proceedings of the 24th International Conference on World Wide Web Companion, pages 121–122. International World Wide Web Conferences Steering Committee, 2015.

[23] L. Sterckx, T. Demeester, J. Deleu, and C. Develder. *When Topic Models Disagree: Keyphrase Extraction with Multiple Topic Models*. In Proceedings of the 24th International Conference on World Wide Web Companion, pages 123–124. International World Wide Web Conferences Steering Committee, 2015.

[24] G. Salton and C. Buckley. *Term-weighting approaches in automatic text retrieval*. Information processing & management, 24(5):513–523, 1988.

[25] Y. Zhang, N. Zincir-Heywood, and E. Milios. *Narrative Text Classification for Automatic Key Phrase Extraction in Web Document Corpora*. In Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, WIDM '05, pages 51–58, New York, NY, USA, 2005. ACM. Available from: http://doi.acm.org/10.1145/1097047.1097059, doi:10.1145/1097047.1097059.

[26] R. Mihalcea and P. Tarau. *TextRank: Bringing Order into Texts*. In Proceedings of the 2004 conference on EMNLP, 2004. Available from: http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf.

[27] L. Gazendam, C. Wartena, and R. Brussee. *Thesaurus Based Term Ranking for Keyword Extraction*. In Database and Expert Systems Applications, DEXA, International Workshops, Bilbao, Spain, August 30 - September 3, 2010, pages 49–53, 2010. Available from: http://dx.doi.org/10.1109/DEXA.2010.31, doi:10.1109/DEXA.2010.31.

[28] X. Jiang, Y. Hu, and H. Li. *A ranking approach to keyphrase extraction*. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pages 756–757. ACM, 2009.

[29] Y. Park, R. J. Byrd, and B. Boguraev. *Automatic Glossary Extraction: Beyond Terminology Identification*. In 19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002, 2002. Available from: http://aclweb.org/anthology/C02-1142.

[30] L. Gazendam, C. Wartena, V. MalaisÃl', G. Schreiber, A. de Jong, and H. Brugman. *Automatic Annotation Suggestions for Audiovisual Archives: Evaluation Aspects*. Interdisciplinary Science Reviews, 34(2-3):172–188, 2009. Available from: http://dx.doi.org/10.1179/174327909X441090,

arXiv:http://dx.doi.org/10.1179/174327909X441090, doi:10.1179/174327909X441090.

[31] I. Witten, G. Paynter, and E. Frank. *KEA: Practical automatic keyphrase extraction*. Proceedings of the fourth ACM conference on Digital libraries, 1999. Available from: http://dl.acm.org/citation.cfm?id=313437.

[32] D. H. Wolpert and W. G. Macready. *No free lunch theorems for optimization*. IEEE transactions on evolutionary computation, 1(1):67–82, 1997.

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan. *Latent Dirichlet Allocation*. JMLR, 3(4-5):993–1022, 2003. doi:10.1162/jmlr.2003.3.4-5.993.

[34] C.-C. Chang and C.-J. Lin. *LIBSVM: A Library for Support Vector Machines*. ACM Trans. Intell. Syst. Technol., 2(3):27:1–27:27, May 2011.

[35] T. Chen and C. Guestrin. *XGBoost: A Scalable Tree Boosting System*. CoRR, abs/1603.02754, 2016. Available from: http://arxiv.org/abs/1603.02754.

[36] E. Brill. *A simple rule-based part of speech tagger*. In Proceedings of the workshop on Speech and Natural Language, pages 112–116. Association for Computational Linguistics, 1992.

[37] S. Bird. *NLTK: the natural language toolkit*. In Proceedings of the COLING/ACL on Interactive presentation sessions, pages 69–72. Association for Computational Linguistics, 2006.

[38] C. Elkan and K. Noto. *Learning classifiers from only positive and unlabeled data*. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008, pages 213–220, 2008. Available from: http://doi.acm.org/10.1145/1401890.1401920, doi:10.1145/1401890.1401920.

[39] L. Sterckx, C. Caragea, T. Demeester, and C. Develder. *Supervised Keyphrase Extraction as Positive Unlabeled Learning*. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, November 2-4, 2016, Austin, Texas, 2016.

[40] P. Turney. *Learning to extract keyphrases from text*. 1999. Available from: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=8913245.

[41] A. Bougouin and F. Boudin. *TopicRank : ordonnancement de sujets pour lâĂŹextraction automatique de termes-clÃl's*. TAL, 55(1):45–69, 2014.

[42] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin. *Automatic keyphrase extraction from scientific articles*. Language Resources and Evaluation, 47(3):723–742, December 2012. Available from: http://link.springer.com/10.1007/s10579-012-9210-3, doi:10.1007/s10579-012-9210-3.

[43] C. Caragea, F. A. Bulgarov, A. Godea, and S. Das Gollapalli. *Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1435–1446, Doha, Qatar, October 2014. Association for Computational Linguistics. Available from: http://www.aclweb.org/anthology/D14-1150.

[44] A. Hulth. *Improved automatic keyword extraction given more linguistic knowledge*. In Proceedings of the 2003 conference on Empirical methods in natural language processing, pages 216–223, 2003.

[45] L. Marujo, A. Gershman, J. Carbonell, R. Frederking, and J. ao P. Neto. *Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization*. In Proceedings of LREC 2012. ELRA, 2012.

[46] T. D. Nguyen and M.-Y. Kan. *Key phrase Extraction in Scientific Publications*. In Proceeding of International Conference on Asian Digital Libraries, pages 317–326, 2007.

[47] Y. Ren, D. Ji, and H. Zhang. *Positive Unlabeled Learning for Deceptive Reviews Detection*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 488–498, 2014. Available from: http://aclweb.org/anthology/D/D14/D14-1055.pdf.

[48] M. C. du Plessis, G. Niu, and M. Sugiyama. *Analysis of Learning from Positive and Unlabeled Data*. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 703–711, 2014. Available from: http://papers.nips.cc/paper/5509-analysis-of-learning-from-positive-and-unlabeled-data.

[49] E. Bengtson and D. Roth. *Understanding the Value of Features for Coreference Resolution*. In 2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 294–303, 2008. Available from: http://www.aclweb.org/anthology/D08-1031.

[50] K. S. Hasan and V. Ng. *Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-art*. In Proceedings of the 23rd COLING, COLING 2010, pages 365–373, Stroudsburg, PA, USA, 2010. Available from: http://dl.acm.org/citation.cfm?id=1944566.1944608.

[51] V. Hoste. *Optimization issues in machine learning of coreference resolution*. PhD thesis, Universiteit Antwerpen. Faculteit Letteren en Wijsbegeerte., 2005.

5

Sequence-to-Sequence Applications using Weak Supervision

This chapter presents two research papers situated in the domain of sequence-to-sequence models for monolingual data. We introduce these contributions in a seperate introduction section.

⋆⋆⋆

5.1 Introduction

Sequence-to-sequence (seq2seq) models transform sequences of symbols from a source domain (e.g., a particular language) to sequences of symbols in a target domain (e.g., another language). Seq2seq models [1, 2] are one of the most successful applications of neural network architectures to natural language processing. However, these architectures, mostly relying on recurrent neural networks, are heavily parameterized and require large amounts of high-quality training data.

The most common seq2seq model, as introduced by Bahdanau et al. [2], consists of two recurrent neural networks (RNNs, commonly stacked with LSTM memory cells): an encoder which processes the input and a decoder which generates the output. Recently, such RNN-based seq2seq models have been used with success in various natural language processing tasks,

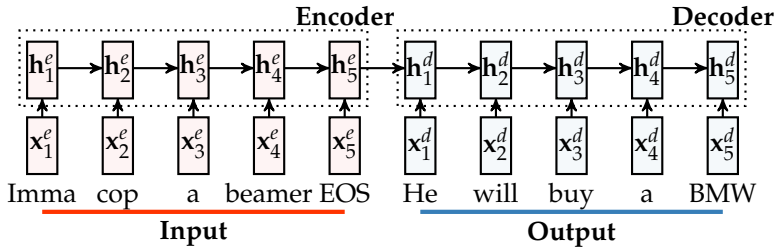


Figure 5.1: Sequence-to-Sequence modeling using recurrent neural networks.

including constituency parsing [3], and, most notably, machine translation [1, 4].

The probability of each output sequence $(y_1, \ldots, y_{T'})$ is conditioned on the corresponding input sequence $(x_1, \ldots, x_T)$, and is modeled as

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1}), \tag{5.1}$$

in which v represents the input sequence. During training, the log probability of an output sequence $\mathbf{y}$ given the considered input sequence $\mathbf{x}$ is maximized over the training set S. The training objective becomes

$$\frac{1}{|S|} \sum_{(\mathbf{x},\mathbf{y}) \in S} \log p(\mathbf{y}|\mathbf{x}). \tag{5.2}$$

Each contribution $p(y_t | v, y_1, \ldots, y_{t-1})$ in the conditional output distribution of each training instance in eq. (5.1) is modeled as a categorical distribution over the output vocabulary terms, by performing a Softmax over the decoder's output at position t. Once training is complete, we produce translations and generated by finding the most likely translation according to the LSTM:

$$T = \arg\max p(T|S) \tag{5.3}$$

In machine translation, text in a source language is read by the encoder RNN, and a decoder RNN produces the translated sentence. NMT is appealing since it requires minimal domain knowledge and is conceptually simple. Apart from its application to machine translation, the seq2seq paradigm has been successfully applied to monolingual text-to-text operations including text simplification [5], paraphrasing [6], style transfer [7], sarcasm interpretation [8], and dialogue systems [9]. Figure 5.1 shows a schematic representation of a seq2seq model for automated lyric annotation. However, there are important differences between both problems, such as the fact that the alignment between input and output sequences is

much weaker or often non-existent. While these tasks are gaining in popularity and more datasets are being released, sizes are often not sufficient for effectively training seq2seq models.

In this chapter we present a novel application of seq2seq with a large accompanying dataset and study the task of text simplification for which only noisy or low quality training data is available. In Section 5.2, we present a novel application called Automated Lyric Annotation. We create a dataset based on annotations generated by users of *Genius.com*, an online lyrics database which provide explanations to lyrics and poetic text. We compare seq2seq models to a retrieval approach and statistical machine translation baseline and show we are able to provide explanations to poetic text as evaluated by mechanical turkers. The created dataset is one of the largest of its kind and we hope it stimulates research in text normalization for social media text, metaphor processing and paraphrase generation.

In Section 5.3, we propose a task agnostic method to boost performance on these tasks using large parallel text collections with no consistent operation going from source to text. We show how attention, generated prior to translation using a generative model, can be used to steer output towards having certain attributes. We apply this technique for the task of text simplification. Text simplification transforms text into a more simple and direct style, using a smaller vocabulary that substitutes infrequent and otherwise difficult words (such as long composite nouns, technical terms, neologisms and abstract concepts) by simpler corresponding expressions.

5.2 Break it Down for Me: A Study in Automated Lyric Annotation

This paper presents Automated Lyric Annotation. We create a dataset based on crowdsourced annotations by users of the Genius.com online lyrics database which allows users to provide explanations to lyrics and poetic text. Our seq2seq model is able to provide explanations to held-out poetic text. The created dataset is one of the largest of its kind and stimulates research in text normalization for social media text, metaphor processing and paraphrase generation.

⋆ ⋆ ⋆

L. Sterckx, J. Naradowsky, B. Byrne, T. Demeester and C. Develder

Presented at the Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017.

Abstract Comprehending lyrics, as found in songs and poems, can pose a challenge to human and machine readers alike. This motivates the need for systems that can understand the ambiguity and jargon found in such creative texts, and provide commentary to aid readers in reaching the correct interpretation. We introduce the task of automated lyric annotation (ALA). Like text simplification, a goal of ALA is to rephrase the original text in a more easily understandable manner. However, in ALA a system should often include *new or additional* information to clarify niche terminology and abstract concepts. To stimulate research on this task, we release a large collection of crowdsourced annotations for song lyrics. We analyze the performance of translation and retrieval models on this task, measuring performance with both automated and human evaluation. We find that each model captures a unique type of information important to the task.

5.2.1 Introduction

Song lyrics and poetry often make use of ambiguity, symbolism, irony, and other stylistic elements to evoke emotive responses. These characteristics sometimes make it challenging to interpret obscure lyrics, especially for readers or listeners who are unfamiliar with the genre. To address this problem, several online lyric databases have been created where users can explain, contextualize, or discuss lyrics. Examples include MetroLyrics[1] and Genius.com[2]. We refer to such commentary as a lyric annotation (Figure 5.2).

[1] http://www.metrolyrics.com
[2] http://genius.com

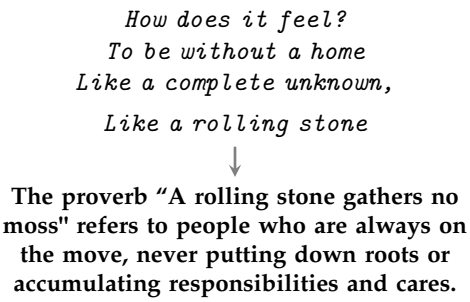


Figure 5.2: A lyric annotation for “Like A Rolling Stone" by Bob Dylan.

In this work we introduce the task of *automated lyric annotation* (ALA). Compared to many traditional NLP systems, which are trained on newswire or similar text, an automated system capable of explaining abstract language, or finding alternative text expressions for slang (and other unknown terms) would exhibit a deeper understanding of the nuances of language. As a result, research in this area may open the door to a variety of interesting use cases. In addition to providing lyric annotations, such systems can lead to improved NLP analysis of informal text (blogs, social media, novels and other literary works of fiction), better handling of genres with heavy use of jargon (scientific texts, product manuals), and increased robustness to textual variety in more traditional NLP tasks and genres.

Our contributions are as follows:

1. To aid in the study of ALA we present a corpus of 803,720 crowdsourced lyric annotation pairs suitable for training models for this task.
2. We present baseline systems using statistical machine translation (SMT), neural translation (Seq2Seq), and information retrieval.
3. We establish an evaluation procedure which adopts measures from machine translation, paraphrase generation, and text simplification. Evaluation is conducted using both human and automated means, which we perform and report across all baselines.

5.2.2 The Genius ALA Dataset

We collect a dataset of crowdsourced annotations, generated by users of the *Genius* online lyric database. For a given song, users can navigate to a particular stanza or line, view existing annotations for the target lyric, or provide their own annotation. Discussion between users acts to improve annotation quality, as it does with other collaborative online databases like Wikipedia. This process is gamified: users earn *IQ* points for producing high quality annotations.

We collect 736,423 lyrics having a total 1,404,107 lyric annotation pairs

Table 5.1: Properties of gathered dataset (V_{lyrics} and V_{annot} denote the vocabulary for lyrics and annotations, $\oslash$ denotes the average amount).

# Lyric Annotation pairs	803,720
$\oslash$ Tokens per Lyric	15
$\oslash$ Tokens per Annotation	43
$\lvert V_{\text{lyrics}} \rvert$	124,022
$\lvert V_{\text{annot}} \rvert$	260,427

from all subsections (rap, poetry, news, etc.) of Genius. We limit the initial release of the annotation data to be English-only, and filter out non-English annotations using a pre-trained language identifier. We also remove annotations which are solely links to external resources, and do not provide useful textual annotations. This reduces the dataset to 803,720 lyric annotation pairs. We list several properties of the collected dataset in Table 5.1.

5.2.3 Context Independent Annotation

Mining annotations from a collaborative human-curated website presents additional challenges worth noting. For instance, while we are able to generate large quantities of parallel text from Genius, users operate without a single, predefined and shared *global* goal other than to maximize their own IQ points. As such, there is no motivation to provide annotations for a song in its entirety, or independent of previous annotations.

For this reason we distinguish between two types of annotations: *context independent* (CI) annotations are independent of their surrounding context and can be interpreted without it, e.g., explain specific metaphors or imagery or provide narrative while normalizing slang language. Contrastively, *context sensitive* (CS) annotations provide broader context beyond the song lyric excerpt, e.g., background information on the artist.

To estimate contribution from both types to the dataset, we sample 2,000 lyric annotation pairs and label them as either CI or CS. Based on this sample, an estimated 34.8% of all annotations is independent of context. Table 5.2 shows examples of both types.

While the goal of ALA is to generate annotations of all types, it is evident from our analysis that CS annotations can not be generated by models trained solely on parallel text. That is, these annotations cannot be generated without background knowledge or added context. Therefore, in this preliminary work we focus on predicting CI lyric annotations.

Table 5.2: Examples of context independent and dependent pairs of lyrics [L] and annotations [A].

Type	% of **annotations**	**Examples**	
CI	34.8%	[L]	`Gotta patch a lil kid tryna get at this cabbage`
(Context		[A]	He's trying to ignore the people trying to get at his money.
independent)		[L]	`You know it's beef when a smart brother gets stupid`
		[A]	You know an argument is serious when an otherwise rational man loses rational.
CS	65.2%	[L]	`Cause we ain't break up, more like broke down`
(Context		[A]	The song details Joe's break up with former girlfriend Esther.
sensitive)		[L]	`If I quit this season, I still be the greatest, funk`
		[A]	Kendrick has dropped two classic albums and pushed the artistic envelope further.

5.2.4 Baselines

We experiment with three baseline models used for text simplification and paraphrase generation.

- **Statistical Machine Translation (SMT):** One approach is to treat the task as one of translation, and to use established statistical machine translation (SMT) methods [10] to produce them. We train a standard phrase-based SMT model to translate lyrics to annotations, using GIZA++ [11] for word alignment and Moses [12] for phrasal alignment, training, and decoding.

- **Seq2Seq:** Sequence-to-sequence models [1] offer an alternative to SMT systems, and have been applied successfully to a variety of tasks including machine translation. In Seq2Seq, a recurrent neural network (RNN) encodes the source sequence to a single vector representation. A separate decoder RNN generates the translation conditioned on this representation of the source sequence's semantics. We utilize Seq2Seq with attention [2], which allows the model to additionally condition on tokens from the input sequence during decoding.

- **Retrieval:** In practice, similar lyrics may reappear in different contexts with exchangeable annotations. We treat the training corpus as a database of lyrics' excerpts with corresponding annotations, and at test time select the annotation assigned to the most similar lyric. This

Table 5.3: Lyrics excerpts with annotations from Genius ('Human') and automated annotators.

Lyric:	`and when I slept, I dream g's, son I need some`
Human:	he dreams about large amounts of money because he is in need of it .
SMT:	and when he slept, and he's dream, g's son, he needs some
Seq2Seq:	when he sleeps , he dreams of getting money.
Retrieval:	g's refer to the ghetto gangsters in london that the kid has become involved with
Lyric:	`Mossberg pump I'm ridin' shotgun literally`
Human:	riding shotgun means to ride up front in the passenger seat
SMT:	Mossberg pump - action shotgun, he's saying he's ridin' shotgun literally.
Seq2Seq:	Mossberg is a firearms manufacturer. He is riding shotgun with his shotgun.
Retrieval:	he fully respects the people mentioned earlier and is riding for them.
Lyric:	`and my eyes red cause of all that haze`
Human:	his eyes turn red due to lack of hydration from the weed one smokes (in this case , smoking purple haze)
SMT:	and his eyes are red because of all that haze.
Seq2Seq:	his eyes are red because of all the weed he smokes.
Retrieval:	his eyes are red as a sign of the fatigue of travel and , possibly , from drugs and alcohol .

baseline is referred to as the *retrieval* model. We use standard TF-IDF weighted cosine distance as similarity measure between lyrics' excerpts.

5.2.5 Evaluation

5.2.5.1 Data

We evaluate automatic annotators on a selection of 354 CI annotations and partition the rest of the annotations into 2,000 instances for development and the full remainder for training. It is important to note that the annotations used for training and development include CI as well as CS annotations.

Annotations often include multiple sentences or even paragraphs for a single lyrics excerpt (which does not include end marks), while machine translation models need aligned corpora at sentence level to perform well [13]. We therefore transform training data by including each sentence from the annotation as a single training instance with the same lyric, resulting in a total of 1,813,350 sentence pairs.

We use this collection of sentence pairs (denoted as *sent.* in results) to train the SMT model. Seq2Seq models are trained using sentence pairs as

well as full annotations. Interestingly, techniques encouraging alignment by matching length and thresholding cosine distance between lyric and annotation did not improve performance during development.

5.2.6 Measures

For automated evaluation, we use measures commonly used to evaluate translation systems (BLEU, METEOR), paraphrase generation (iBLEU) and text simplification (SARI).

BLEU [14] uses a modified form of precision to compare generated annotations against references from Genius. METEOR [15] is based on the harmonic mean of precision and recall and, along with exact word matching, includes stemming and synonymy matching. iBLEU [16] is an extension of the BLEU metric to measure diversity as well as adequacy of the annotation, iBLEU $= 0.9 \times$ BLEU(Annotation, Reference) $- 0.1 \times$ BLEU(Annotation, Lyric). SARI [13] measures precision and recall of words that are added, kept, or deleted separately and averages their arithmetic means.

We also measure quality by crowdsourcing ratings via the online platform CrowdFlower.[3] We present collaborators with a song lyric excerpt annotated with output from the annotation generators as well as a reference annotation from Genius. Collaborators assign a 5-point rating for *Fluency* which rates the quality of the generated language, and *Information* which measures the added clarification by the annotation, a key aspect of this task. For each lyric annotation pair, we gather ratings from three different collaborators and take the average.

5.2.6.1 Hyperparameters and Optimization

Here we describe implementation and some of the optimizations used when training the models. For seq2seq models, we use OpenNMT [17] and optimize for perplexity on the development set. Vocabulary for both lyrics and annotations is reduced to the 50,000 most frequent tokens and are embedded in a 500-dimensional space. We use two layers of stacked bidirectional LSTMs with hidden states of 1024 dimensions. We regularize using dropout (keep probability of 0.7) and train using stochastic gradient descent with batches of 64 samples for 13 epochs. The decoder of the SMT model is tuned for optimal BLEU scores on the development set using minimum error rate training [18].

[3] https://www.crowdflower.com/

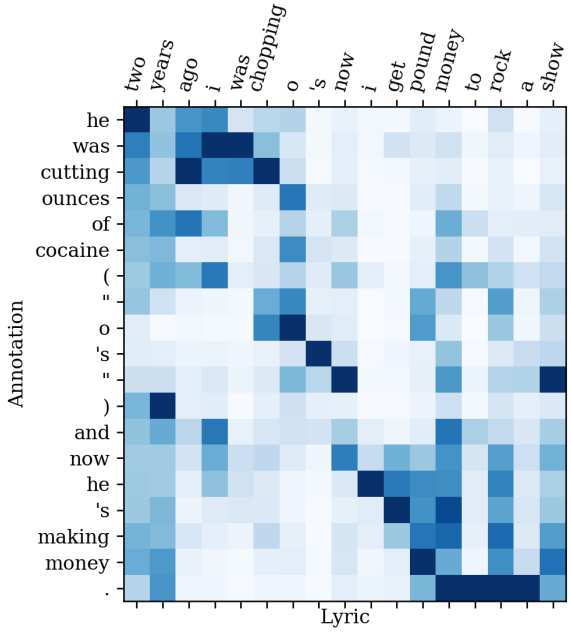


Figure 5.3: Attention visualization of seq2seq models for ALA.

5.2.7 Results

To measure agreement between collaborators, we compute the kappa statistic [19]. Kappa statistics for fluency and information are 0.05 and 0.07 respectively, which indicates low agreement. The task of evaluating lyric annotations was difficult for CrowdFlower collaborators as was apparent from their evaluation of the task. For evaluation in future work, we recommend recruitment of expert collaborators familiar with the Genius platform and song lyrics. Table 5.3 shows examples of lyrics with annotations from Genius and those generated by baseline models. A notable observation is that translation models learn to take the role of narrator, as is common in CI annotations, and recognize slang language while simplifying it to more standard English.

Automatic and human evaluation scores are shown in Table 5.6. Next to evaluation metrics, we show two properties of automatically generated annotations; the average annotation length relative to the lyric and the occurrence of profanity per token in annotations, using a list of 343 swear words.

The SMT model scores high on BLEU, METEOR and SARI but shows a large drop in performance for iBLEU, which penalizes lexical similarity between lyrics and generated annotations as apparent from the amount profanity remaining in the generated annotations.

Standard SMT rephrases the song lyric from a third person perspective but is conservative in lexical substitutions and keeps close to the grammar of the lyric. A more appropriate objective function for tuning the decoder which promotes lexical dissimilarity as done for paraphrase generation, would be beneficial for this approach.

Seq2seq models generate annotations more dissimilar to the song lyric and obtain higher iBLEU and Information scores. To visualize some of the alignments learned by the translation models, Fig. 5.3 shows word-by-word attention scores for a translation by the seq2seq model.

While the retrieval model obtains quality annotations when test lyrics are highly similar to lyrics from the training set, retrieved annotations are often unrelated to the test lyric or specific to the song lyric it is retrieved from.

Out of the unsupervised metrics, METEOR obtained the highest Pearson correlation [20] with human ratings for Information with a coefficient of 0.15.

5.2.8 Related Work

Work on modeling of social annotations has mainly focused on the use of topic models [21, 22] in which annotations are assumed to originate from topics. They can be used as a preprocessing step in machine learning tasks such as text classification and image recognition but do not generate language as required in our ALA task.

Text simplification and paraphrase generation have been widely studied. Recent work has highlighted the need for large text collections [23] as well as more appropriate evaluation measures [13, 24]. They indicated that especially informal language, with its high degree of lexical variation, e.g., as used in social media or lyrics, poses serious challenges [25].

Text generation for artistic purposes, such as poetry and lyrics, has been explored most commonly using templates and constraints [26]. In regard to rap lyrics, Wu et al. [27] present a system for rap lyric generation that produces a single line of lyrics that is meant to be a response to a single line of input. Most recent work is that of Zhang et al. [28] and Potash et al. [29], who show the effectiveness of RNNs for the generation of poetry and lyrics.

The task of annotating song lyrics is also related to metaphor processing. As annotators often explain metaphors used in song lyrics, the Genius dataset can serve as a resource to study computational modeling of metaphors [30].

5.2.9 Conclusion and Future Work

We presented and released the Genius dataset to study the task of Automated Lyric Annotation. As a first investigation, we studied automatic

generation of context independent annotations as machine translation and information retrieval. Our baseline system tests indicate that our corpus is suitable to train machine translation systems.

Standard SMT models are capable of rephrasing and simplifying song lyrics but tend to keep close to the structure of the song lyric. Seq2Seq models demonstrated potential to generate more fluent and informative text, dissimilar to the lyric.

A large fraction of the annotations is heavily based on context and background knowledge (CS), one of their most appealing aspects. As future work we suggest injection of structured and unstructured external knowledge [31] and explicit modeling of references [32].

	Properties		Automated Evaluation				Human Evaluation	
	Len. Ratio	Profanity	BLEU	iBLEU	METEOR	SARI	Fluency	Info.
Human	1.19	0.0027	-	-	-	-	3.93	3.53
SMT (Sent.)	1.23	0.0068	6.22	1.44	12.20	38.42	3.82	3.31
Seq2Seq (Sent.)	1.05	0.0023	5.33	3.64	9.28	36.52	3.76	3.25
Seq2Seq	1.32	0.0022	5.15	3.46	10.56	36.86	3.83	3.34
Retrieval	1.18	0.0038	2.82	2.27	5.10	32.76	3.93	2.98

Table 5.4: Quantitative evaluation of different automated annotators.

5.3 Prior Attention for Style-aware Sequence-to-Sequence Models

This paper proposes a task agnostic method to boost performance on seq2seq tasks using large parallel text collections with no consistency in operations performed when going from source to text. We show how attention, generated prior to translation using a generative model, can be used to steer output towards having certain stylistic attributes.

⋆ ⋆ ⋆

L. Sterckx, J. Deleu, C. Develder and T. Demeester

Submitted to Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018.

Abstract We extend sequence-to-sequence models with the possibility to control the characteristics or style of the generated output, via attention that is generated a priori (before decoding) from a latent code vector. After training an initial attention-based sequence-to-sequence model, we use a variational auto-encoder conditioned on representations of input sequences and a latent code vector space to generate attention matrices. By sampling the code vector from specific regions of this latent space during decoding and imposing prior attention generated from it in the seq2seq model, output can be steered towards having certain attributes. This is demonstrated for the task of sentence simplification, where the latent code vector allows control over output length and lexical simplification, and enables fine-tuning to optimize for different evaluation metrics.

5.3.1 Introduction

Apart from its application to machine translation, the *encoder-decoder* or *sequence-to-sequence* (seq2seq) paradigm has been successfully applied to monolingual text-to-text tasks including simplification [5], paraphrasing [6], style transfer [7], sarcasm interpretation [8], automated lyric annotation [33] and dialogue systems [9]. A sequence of input tokens is encoded to a series of hidden states using an encoder network and decoded to a target domain by a decoder network. During decoding, an attention mechanism is used to indicate which are the relevant input tokens at each step. This attention component is computed as an intermediate part of the model, and is trained jointly with the rest of the model.

Alongside being crucial for effective translation, attention — while not necessarily correlated with human attention — brings interpretability to seq2seq models by visualizing how individual input elements contribute to the model's decisions. Attention values typically match up well with

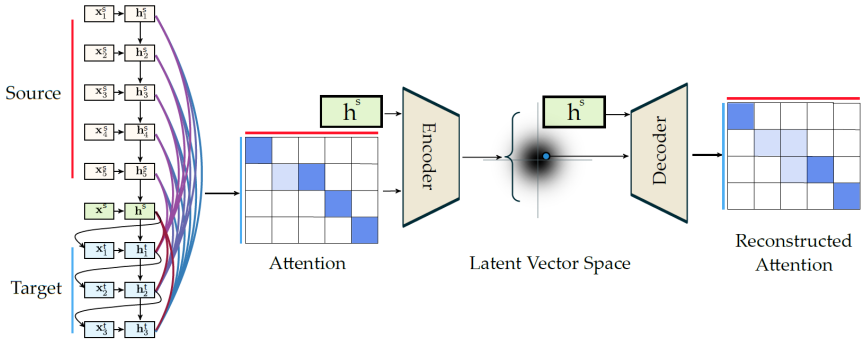


Figure 5.4: Training of a conditional variational autoencoder applied to attention matrices. The seq2seq model translates training sentences from the source to a target domain while generating attention matrices. These matrices are concatenated with a representation of the source sentence and encoded to a low dimensional latent vector space.

word alignments used in traditional statistical machine translation, obtained with tools such as GIZA++ [34] or fast-align [35]. Therefore, several works have included prior alignments from dedicated alignment software such as GIZA++ or fast-align [36–38]. In particular, [37] showed that the distance between the attention-infused alignments and the ones learned by an independent alignment model can be added to the networks' training objective, resulting in improved translation and alignment quality. Further, [39] demonstrated that this alignment between given input sentence and generated output can be planned ahead as part of a seq2seq model: their model makes a plan of future alignments using an alignment-plan matrix and decides when to follow this plan by learning a separate commitment vector. In the standard seq2seq model, where attention is calculated at each time step, such overall alignment or focus is only apparent after decoding and is thus not carefully planned nor controlled. We hypothesize that many text-to-text operations have varying levels of alignment and focus. To enable control over these aspects, we propose to precompute alignments and use this *prior* attention to determine the structure or focus before decoding in order to steer output towards having specific attributes, such as length or level of compression.

We facilitate this control through an input represented in a latent vector space (rather than, e.g., explicit 'style' attributes).

After training of the initial seq2seq model (with standard attention) on a parallel text corpus, a conditional variational autoencoder [40] learns to reconstruct matrices of alignment scores or attention matrices from a latent vector space and the input sentence encoding. At translation time, we are able to efficiently generate specific attention by sampling from regions in

the latent vector space, resulting in output having specific stylistic attributes. We apply this method on a sentence simplification corpus, showing that we can control length and compression of output while producing realistic output and allowing fine-tuning for optimal evaluation scores.

5.3.2 Generation of Prior Attention

This section describes our proposed method, sketched in Figure 5.4, with emphasis on the generation of prior attention matrices.

An encoder recurrent neural network computes a sequence of representations over the source sequence, i.e., its hidden states $\mathbf{h}_i^s$ (with $i = 1, \ldots, n$ and n the length of the source sequence). In attention-based models, an alignment vector $\mathbf{a}_j = [\alpha_{j,1}, ..., \alpha_{j,n}]$ is obtained by comparing the current target hidden state $\mathbf{h}_j^t$ with each source hidden state $\mathbf{h}_i^s$. A global context vector $\mathbf{c}_j$ is then computed as the weighted average, according to alignment weights of $\mathbf{a}_j$, over all the source states $\mathbf{h}_i^s$ at time step j (for $j = 1, ..., m$ over m decoding steps). After decoding, these alignment vectors form a matrix $\mathbf{A}$ of attention vectors, $\mathbf{A} = [\mathbf{a}_1; \mathbf{a}_2; \ldots; \mathbf{a}_m]$ capturing the alignment between source and target sequence.

Table 5.5: Output excerpts for prior attention matrices sampled from a 2D latent vector space. Samples are drawn from outer regions, with + indicating large positive values and − for negative values.

z_1	z_2	
		The wave traveled across the Atlantic , and organized into a tropical depression off the northern coast of Haiti on September 13 .
-	-	The wave traveled across the Atlantic , and organized into a tropical depression off the northern coast of the country on September 13 .
-	+	The wave traveled across the Atlantic Ocean into a tropical depression off the northern coast of Haiti on September 13 .
+	-	The wave traveled across the Atlantic Ocean and the Pacific Ocean to the south , and the Pacific Ocean to the south , and the Atlantic Ocean to the west .
+	+	The storm was the second largest in the Atlantic Ocean .
z_1	z_2	**Below are some useful links to facilitate your involvement .**
-	-	Below are some useful links to facilitate your involvement .
-	+	Below are some useful links to help your involvement .
+	-	Below are some useful to be able to help help develop to help develop .
+	+	Below is a software program that is used to talk about what is now .

Inspired by the field of image generation, we treat alignment matrices as grayscale images and use generative models to create previously unseen attention. Generative models have been applied to a variety of problems giving state-of-the-art results in image generation, text-to-speech synthesis, and image captioning. One of the most prominent models is the variational autoencoder (VAE) proposed by [41]. Given an observed variable $\mathbf{x}$, the VAE introduces a continuous latent variable $\mathbf{z}$, and assumes $\mathbf{x}$ to be generated from $\mathbf{z}$, i.e., $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, with $p(\mathbf{z})$ being a prior over

the latent variables. $p_D(\mathbf{x}|\mathbf{z})$ is the conditional distribution that models the generation procedure parameterized by a decoder network D. For a given $\mathbf{x}$, an encoder network E outputs a variational approximation $q_E(\mathbf{z}|\mathbf{x})$ of the true posterior over the latent values $p(\mathbf{z}|\mathbf{x}) \propto p_D(\mathbf{x}|\mathbf{z})p_Z(\mathbf{z})$. The parameters of E, D are learned using stochastic variational inference to maximize a lower bound for the marginal likelihood of each observation in the training data. In our setting, $\mathbf{x}$ represents the attention matrix.

Next to control over stylistic features, we want attention matrices to be relevant for a specific source sentence. In the Conditional Variational Autoencoder (CVAE) [40, 42], the standard VAE is conditioned on additional variables which can be used to generate diverse images conditioned on certain attributes, e.g., generating different human faces given a sentiment. We view the source contexts as the added conditional attributes and use the CVAE to generate diverse attention matrices instead of images. This context vector is represented by the source sentence encoding $\mathbf{h}^s$. The CVAE encoder is conditioned on two variables, the attention matrix $\mathbf{A}$ and the sentence encoding $q_E(\mathbf{z}|\mathbf{A}, \mathbf{h}^s)$. Analogous for the decoder, the likelihood is now conditioned on two variables, a latent code $\mathbf{z}$ and again the source sentence encoding, $p_D(\mathbf{A}|\mathbf{z}, \mathbf{h}^s)$. The variational lower bound objective becomes

$$\mathcal{L}(E, D, c) = \mathbb{E}_{z\sim q_E}[\log p_D(\mathbf{X}|\mathbf{z}, c)] - \mathrm{D}_{\mathrm{KL}}[q_E(\mathbf{z}|\mathbf{X}, c)\|p(\mathbf{z}|c)] \tag{5.4}$$

i.e. we condition distributions on the additional variable $c = h^s$ denoting the input sentence. This training procedure is visualized in Figure 5.4. This training procedure of the CVAE is visualized in Figure 5.4. At test time, the attention scores from the attention matrix, pre-generated from a latent code sample and the source sentence encoding, are used instead of the standard seq2seq model's attention mechanism.

Table 5.6: Quantitative evaluation of existing baselines and seq2seq with prior attention from the CVAE when choosing an optimal z sample for BLEU scores. For comparison, we include the *NTS* model from [5] and the *EncDecA* by [44].

	BLEU	SARI	Length	FKGL
[43]	67.74	35.34	0.90	10.0
[44]	90.00	37.62	0.95	10.4
[5]	88.16	33.86	0.91	10.1
Seq2seq + attention	89.92	33.06	0.91	10.3
Seq2seq + CVAE	90.14	38.30	0.97	10.5

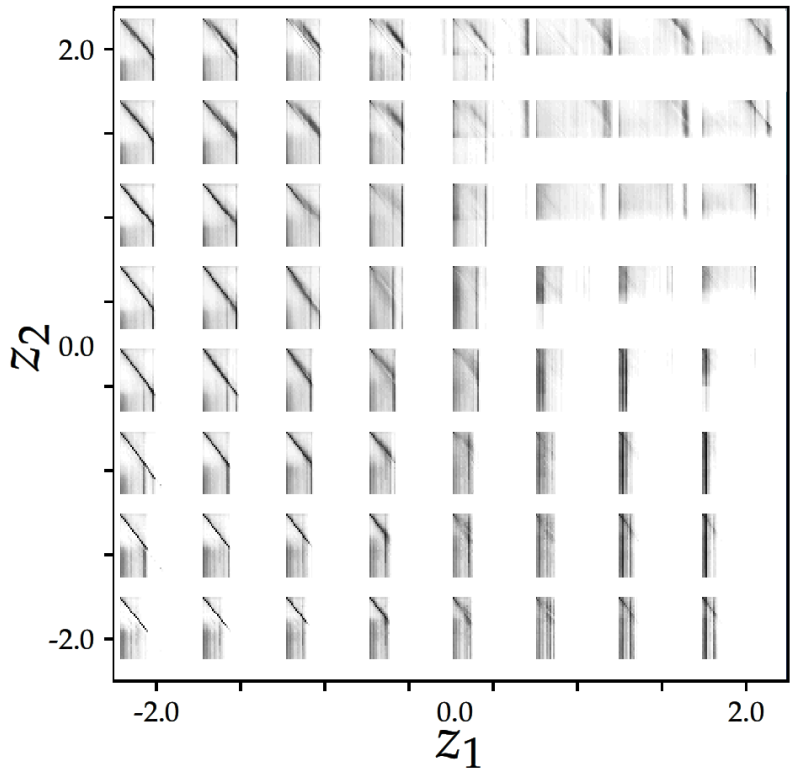


Figure 5.5: (a) Attention matrices for a single source sentence encoding and a two-dimensional latent vector space. By conditioning the autoencoder on the source sentence, the decoder recognizes the length of the source and reduces attention beyond the last source token.

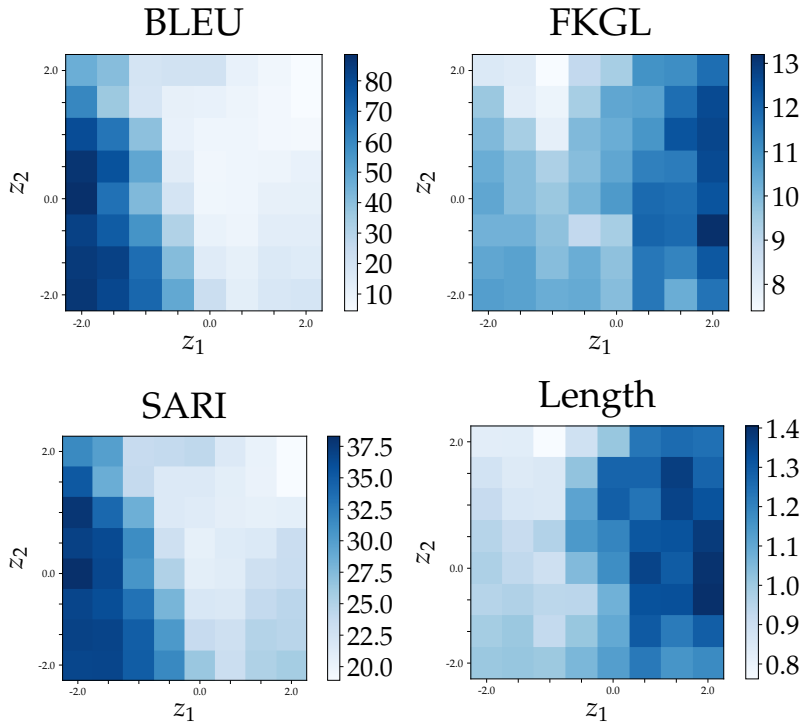


Figure 5.6: (b) Score distributions for different regions of the latent vector space.

5.3.3 Experiments

5.3.3.1 Prior Attention for Text Simplification

While our model is essentially task-agnostic, we demonstrate prior attention for the task of sentence simplification. The goal of sentence simplification is to reduce the linguistic complexity of text, while still retaining its original information and meaning. It has been suggested that sentence simplification can be defined by three major types of operations: *splitting*, *deletion*, and *paraphrasing* [45]. We hypothesize that these operations occur at varying frequencies in the training data. We adopt our model in an attempt to capture these operations into attention matrices and the latent vector space, and thus control the form and degree of simplification through sampling from that space. We train on the *Wikilarge* collection used by Zhu [46]. *Wikilarge* is a collection of 296,402 automatically aligned complex and simple sentences from the ordinary and simple English Wikipedia corpora, used extensively in previous work [5, 43, 44, 47]. The training data includes 2,000 development and 359 test instances created by [13]. These are complex sentences paired with simplifications provided by Amazon Mechanical Turk workers and provide a more reliable evaluation of the task.

5.3.3.2 Hyperparameters and Optimization

We extend the OpenNMT [17] framework with functions for attention generation.We use a similar architecture as [46] and [5]: 2 layers of stacked unidirectional LSTMs with bi-linear global attention as proposed by [48], with hidden states of 512 dimensions. The vocabulary is reduced to the 50,000 most frequent tokens and embedded in a shared 500-dimensional space. We train using SGD with batches of 64 samples for 13 epochs after which the autoencoder is trained by translating sequences from training data. Both the encoder and decoder of the CVAE comprise 2 fully connected layers of 128 nodes. Weights are optimized using ADAM [49]. We visualize and evaluate using a two-dimensional latent vector space. Source and target sequences are both padded or reduced to 50 tokens. The integration of the CVAE is analogous across the family of attention-based seq2seq models, i.e., our approach can be applied more generally with different models or training data.

5.3.3.3 Discussion

To study the influence of sampling from different regions in the latent vector space, we visualize the resulting attention matrices and measure simplification quality using automated metrics. Figure 5.5 shows the two-dimensional latent space for a single source sentence encoding using 64 samples ranging from values -2 to 2. Next to the target-to-source length

ratio, we apply automated measures commonly used to evaluate simplification systems [47, 50]: BLEU, SARI [13], FKGL[4] [51]. Automated evaluation metrics for matrices originating from samples from different regions of latent codes are shown in Figure 5.6. Inclusion of an attention mechanism was instrumental to match existing baselines. Our standard seq2seq model with attention, without prior attention, obtains a score of 89.92 BLEU points, which is close to scores obtained by similar models used in existing work on neural text simplification [5, 44]. In Table 5.6, we compare our seq2seq model with attention and without prior attention. An optimal value for BLEU and SARI scores is found $z = [-2, 0]$ For decreasing values of the first hidden dimension z_1, we observe that attention becomes situated at the diagonal, thus keeping closer to the structure of the source sentence and having one-to-one word alignments. For increasing values of z_1, attention becomes more vertical and focused on single encoder states. This type of attention gives more control to the language model, as exemplified by output samples shown in Table 5.5. Output from this region is far longer and less related to the source sentence.

Influence of the second latent variable z_2 is less apparent from the attention matrices. However, sampling across this dimension shows large effects on evaluation metrics. For decreasing values, output becomes more similar to the source, with higher BLEU as a result. Sampling these values along the zero-axis results in the overall highest BLEU *and* SARI score of 90.14 and 38.30 points respectively, trading similarity for simplification and readability.

5.3.4 Conclusion

We introduced a method to control the decoding process in sequence-to-sequence models using attention, in terms of stylistic characteristics of the output. Given the input sequence and an additional code vector to influence decoding characteristics, a variational autoencoder generates an attention matrix, which is used by the decoder to generate the output sequence according to the alignment style directed by the code vector. We demonstrated the resulting variations in output for the task of text simplification. Yet, our method can be applied to any form of parallel text: we expect different types of training collections, such as translation or style transfer, to give rise to different characteristics or mappings in the latent space.

[4] Fleish-Kincaid Grade Level index.

References

[1] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112, 2014. Available from: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

[2] D. Bahdanau, K. Cho, and Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. CoRR, abs/1409.0473, 2014. Available from: http://arxiv.org/abs/1409.0473.

[3] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. E. Hinton. *Grammar as a Foreign Language*. In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 2773–2781, 2015. Available from: http://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.

[4] T. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba. *Addressing the Rare Word Problem in Neural Machine Translation*. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 11–19. Association for Computational Linguistics, 2015. Available from: http://aclweb.org/anthology/P15-1002, doi:10.3115/v1/P15-1002.

[5] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu. *Exploring Neural Text Simplification Models*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 85–91. Association for Computational Linguistics, 2017. Available from: http://www.aclweb.org/anthology/P17-2014, doi:10.18653/v1/P17-2014.

[6] J. Mallinson, R. Sennrich, and M. Lapata. *Paraphrasing Revisited with Neural Machine Translation*. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 881–893, Valencia, Spain, April 2017. Association for Computational Linguistics. Available from: http://www.aclweb.org/anthology/E17-1083.

[7] H. Jhamtani, V. Gangal, E. Hovy, and E. Nyberg. *Shakespearizing Modern Language Using Copy-Enriched Sequence to Sequence Models*. In Proceedings of the Workshop on Stylistic Variation, pages 10–19, Copenhagen, Denmark, September 2017. Association for Computational

Linguistics. Available from: http://www.aclweb.org/anthology/W17-4902.

[8] L. Peled and R. Reichart. *Sarcasm SIGN: Interpreting Sarcasm with Sentiment Based Monolingual Machine Translation*. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1690–1700, Vancouver, Canada, July 2017. Association for Computational Linguistics. Available from: http://aclweb.org/anthology/P17-1155.

[9] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau. *Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models.* In AAAI, volume 16, pages 3776–3784, 2016.

[10] C. Quirk, C. Brockett, and W. Dolan. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 2004. Available from: http://aclweb.org/anthology/W04-3219.

[11] F. Josef Och and H. Ney. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, Volume 29, Number 1, March 2003, 2003. Available from: http://aclweb.org/anthology/J03-1002.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. *Moses: Open Source Toolkit for Statistical Machine Translation*. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180. Association for Computational Linguistics, 2007. Available from: http://aclweb.org/anthology/P07-2045.

[13] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch. *Optimizing Statistical Machine Translation for Text Simplification*. Transactions of the Association of Computational Linguistics, 4:401–415, 2016. Available from: http://aclweb.org/anthology/Q16-1029.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. *Bleu: a Method for Automatic Evaluation of Machine Translation*. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002. Available from: http://aclweb.org/anthology/P02-1040.

[15] M. Denkowski and A. Lavie. *Proceedings of the Sixth Workshop on Statistical Machine Translation*. pages 85–91, 2011. Available from: http://aclweb.org/anthology/W11-2107.

[16] H. Sun and M. Zhou. *Joint Learning of a Dual SMT System for Paraphrase Generation*. In Proceedings of the 50th Annual Meeting of the

Association for Computational Linguistics (Volume 2: Short Papers), pages 38–42. Association for Computational Linguistics, 2012. Available from: http://aclweb.org/anthology/P12-2008.

[17] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush. *OpenNMT: Open-Source Toolkit for Neural Machine Translation*. ArXiv e-prints, 2017. Available from: https://arxiv.org/abs/1701.02810, arXiv:1701.02810.

[18] N. Bertoldi, B. Haddow, and J. Fouet. *Improved Minimum Error Rate Training in Moses*. Prague Bull. Math. Linguistics, 91:7–16, 2009. Available from: http://ufal.mff.cuni.cz/pbml/91/art-bertoldi.pdf.

[19] J. L. Fleiss. *Measuring nominal scale agreement among many raters.* Psychological bulletin, 76(5):378, 1971.

[20] K. Pearson. *Note on regression and inheritance in the case of two parents*. Proceedings of the Royal Society of London, 58:240–242, 1895.

[21] T. Iwata, T. Yamada, and N. Ueda. *Modeling Social Annotation Data with Content Relevance using a Topic Model*. In Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada., pages 835–843, 2009.

[22] M. K. Das, T. Bansal, and C. Bhattacharyya. *Going beyond Corr-LDA for detecting specific comments on news & blogs*. In Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014, pages 483–492, 2014. Available from: http://doi.acm.org/10.1145/2556195.2556231, doi:10.1145/2556195.2556231.

[23] W. Xu, C. Callison-Burch, and C. Napoles. *Problems in Current Text Simplification Research: New Data Can Help*. Transactions of the Association of Computational Linguistics, 3:283–297, 2015. Available from: http://aclweb.org/anthology/Q15-1021.

[24] M. Galley, C. Brockett, A. Sordoni, Y. Ji, M. Auli, C. Quirk, M. Mitchell, J. Gao, and B. Dolan. *deltaBLEU: A Discriminative Metric for Generation Tasks with Intrinsically Diverse Targets*. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 445–450. Association for Computational Linguistics, 2015. Available from: http://aclweb.org/anthology/P15-2073, doi:10.3115/v1/P15-2073.

[25] W. Xu, A. Ritter, and R. Grishman. *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*. pages 121–128. Association

for Computational Linguistics, 2013. Available from: http://aclweb.org/anthology/W13-2515.

[26] G. Barbieri, F. Pachet, P. Roy, and M. D. Esposti. *Markov Constraints for Generating Lyrics with Style*. In ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012, pages 115–120, 2012. Available from: http://dx.doi.org/10.3233/978-1-61499-098-7-115, doi:10.3233/978-1-61499-098-7-115.

[27] D. Wu, K. Addanki, M. Saers, and M. Beloucif. *Learning to Freestyle: Hip Hop Challenge-Response Induction via Transduction Rule Segmentation*. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 102–112. Association for Computational Linguistics, 2013. Available from: http://aclweb.org/anthology/D13-1011.

[28] X. Zhang and M. Lapata. *Chinese Poetry Generation with Recurrent Neural Networks*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 670–680. Association for Computational Linguistics, 2014. Available from: http://aclweb.org/anthology/D14-1074, doi:10.3115/v1/D14-1074.

[29] P. Potash, A. Romanov, and A. Rumshisky. *GhostWriter: Using an LSTM for Automatic Rap Lyric Generation*. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1919–1924. Association for Computational Linguistics, 2015. Available from: http://aclweb.org/anthology/D15-1221, doi:10.18653/v1/D15-1221.

[30] E. Shutova and S. Teufel. *Metaphor Corpus Annotated for Source - Target Domain Mappings*. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10). European Languages Resources Association (ELRA), 2010. Available from: http://aclweb.org/anthology/L10-1419.

[31] S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio. *A Neural Knowledge Language Model*. ArXiv e-prints, August 2016. Available from: https://arxiv.org/abs/1608.00318, arXiv:1608.00318.

[32] Z. Yang, P. Blunsom, C. Dyer, and W. Ling. *Reference-Aware Language Models*. ArXiv e-prints, November 2016. Available from: https://arxiv.org/abs/1611.01628, arXiv:1611.01628.

[33] L. Sterckx, J. Naradowsky, B. Byrne, T. Demeester, and C. Develder. *Break it Down for Me: A Study in Automated Lyric Annotation*. In Proceedings of the 2017 Conference on Empirical Methods in Natural

Language Processing, pages 2074–2080, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Available from: https://www.aclweb.org/anthology/D17-1220.

[34] F. J. Och and H. Ney. *Improved Statistical Alignment Models*. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 2000. Available from: http://www.aclweb.org/anthology/P00-1056.

[35] C. Dyer, V. Chahuneau, and N. A. Smith. *A Simple, Fast, and Effective Reparameterization of IBM Model 2*. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 644–648. Association for Computational Linguistics, 2013. Available from: http://www.aclweb.org/anthology/N13-1073.

[36] T. Alkhouli, G. Bretschner, J.-T. Peter, M. Hethnawi, A. Guta, and H. Ney. *Alignment-Based Neural Machine Translation*, 2016. Available from: http://www.aclweb.org/anthology/W16-2206, doi:10.18653/v1/W16-2206.

[37] H. Mi, Z. Wang, and A. Ittycheriah. *Supervised Attentions for Neural Machine Translation*. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2283–2288. Association for Computational Linguistics, 2016. Available from: http://www.aclweb.org/anthology/D16-1249, doi:10.18653/v1/D16-1249.

[38] L. Liu, M. Utiyama, A. Finch, and E. Sumita. *Neural Machine Translation with Supervised Attention*. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3093–3102. The COLING 2016 Organizing Committee, 2016. Available from: http://www.aclweb.org/anthology/C16-1291.

[39] C. Gulcehre, F. Dutil, A. Trischler, and Y. Bengio. *Plan, Attend, Generate: Planning for Sequence-to-Sequence Models*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5474–5483. Curran Associates, Inc., 2017.

[40] K. Sohn, H. Lee, and X. Yan. *Learning Structured Output Representation using Deep Conditional Generative Models*. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 3483–3491. Curran Associates, Inc., 2015.

[41] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. CoRR, abs/1312.6114, 2013. Available from: http://arxiv.org/abs/1312.6114, arXiv:1312.6114.

[42] X. Yan, J. Yang, K. Sohn, and H. Lee. *Attribute2Image: Conditional Image Generation from Visual Attributes.* In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, ECCV (4), volume 9908 of *Lecture Notes in Computer Science*, pages 776–791. Springer, 2016. Available from: http://dblp.uni-trier.de/db/conf/eccv/eccv2016-4.html#YanYSL16.

[43] S. Wubben, A. van den Bosch, and E. Krahmer. *Sentence Simplification by Monolingual Machine Translation*. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1015–1024. Association for Computational Linguistics, 2012. Available from: http://aclweb.org/anthology/P12-1107.

[44] X. Zhang and M. Lapata. *Sentence Simplification with Deep Reinforcement Learning*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 584–594. Association for Computational Linguistics, 2017. Available from: http://aclweb.org/anthology/D17-1062.

[45] M. Shardlow. *A Survey of Automated Text Simplification*. International Journal of Advanced Computer Science and Applications, 2014. doi:10.14569/SpecialIssue.2014.040109.

[46] Z. Zhu, D. Bernhard, and I. Gurevych. *A Monolingual Tree-based Translation Model for Sentence Simplification*. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1353–1361. Coling 2010 Organizing Committee, 2010. Available from: http://www.aclweb.org/anthology/C10-1152.

[47] K. Woodsend and M. Lapata. *Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming*. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 409–420. Association for Computational Linguistics, 2011. Available from: http://www.aclweb.org/anthology/D11-1038.

[48] M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. *Multi-task Sequence to Sequence Learning*. CoRR, abs/1511.06114, 2015. Available from: http://arxiv.org/abs/1511.06114, arXiv:1511.06114.

[49] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. CoRR, abs/1412.6980, 2014.

[50] X. Zhang and M. Lapata. *Sentence Simplification with Deep Reinforcement Learning*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 584–594, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Available from: https://www.aclweb.org/anthology/D17-1062.

[51] J. Kincaid. *Derivation of New Readability Formulas: (automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. Research Branch report. Chief of Naval Technical Training, Naval Air Station Memphis, 1975. Available from: https://books.google.be/books?id=4tjroQEACAAJ.

6

Conclusions and Future Research

NLP enables AI systems to interpret human language and perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, and sentiment analysis. Many effective modern NLP systems are built using supervised machine learning methods, which rely on labeled training data. However, the amount of unlabeled linguistic data available to us is much larger and growing much faster than the amount of labeled data. Recent efforts in machine learning have addressed the increasing need for label-efficient machine learning: the ability to learn in complex domains without requiring large quantities of labeled data. This thesis emphasized the use of efficient supervision for a selection of NLP tasks, and presented a variety of techniques which require less supervision while still reaching state-of-the-art performance. In this section we summarize our contributions, provide potential future directions, and discuss the future of weak supervision for NLP.

6.1 Conclusions

As a first application of weak supervision, in Chapter 2 we investigated supervision for knowledge base population systems. Relation extractors are important components of knowledge base population systems, detecting relations occurring between entities. Training relation extractors required the availability of sufficient training data for every relation in the knowledge base's schema. We showed that the amount of manual labeling can be significantly reduced by first applying distant supervision, which gen-

erates training data by aligning large text corpora with existing knowledge bases. However, this typically results in a highly noisy training set, where many training sentences do not express the intended relation. We introduced a techniques called semantic label propagation in which we used low dimensional representations of shortest dependency paths between entities of interest to bootstrap the classifiers. We showed that, with only minutes of labeling per relation we are able to match or improve on accuracy of fully supervised relation extractors. By applying this technique in a participation in the TAC-KBP shared task for knowledge base population systems, we achieved top-ranking submissions. By using more and less noisy training data, our sparse-feature-based linear classifiers were able to obtain higher accuracies than systems using more sophisticated ensembles and deep learning architectures.

In Chapter 3 we developed a more efficient method for the evaluation of topic models by making use of existing labeled text collections. State-of-the-art unsupervised topic models lead to reasonable statistical models of documents but offer no guarantee of creating topics that are interpretable by humans often a full manual evaluation of topics is needed. Proper evaluation required manual supervision which can be costly for large topic models. Instead, we used existing, smaller labeled text collections to provide us with reference concepts and present a new measure for topic quality based on the alignment between these supervised and unsupervised topics. Our proposed measure was shown to correlate better with human evaluation than existing unsupervised measures.

Chapter 4 presented the creation of large corpora of news, sports and fashion articles annotated with keyphrases by a diverse crowd of laymen and professional writers. Prior, there was little consensus on the definition of the task, a lack of large benchmark collections of keyphrase-labeled data, and a lack of overview of the effectiveness of different techniques. Proper evaluation of keyphrase extraction requires large test collections with multiple opinions, which were before, not available for research. We benchmarked existing techniques for supervised and unsupervised keyphrase extraction on the presented corpora. Next to benchmarking existing techniques we studied the influence of overlap in the annotations on the performance metrics. We concluded this chapter by rephrasing the supervised keyphrase extraction problem as positive unlabeled learning in which a binary classifier is learned in a semi-supervised way from only positive keyphrases and unlabeled candidate phrases. We showed that using multiple annotations leads to more robust automatic keyphrase extractors, and proposed reweighting of single annotator labels based on probabilities by a first-stage classifier. This reweighting approach outperforms other single-annotator state-of-the-art automatic keyphrase extractors on different test collections, if we normalize probabilities per document and include co-reference indicators.

In Chapter 5, we presented two applications of sequence-to-sequence

(seq2seq) models trained on noisy or poorly aligned training data. Seq2seq models are one of the most successful applications of deep learning architectures to natural language processing. These architectures, mostly relying on recurrent neural networks are heavily parameterized and require large amounts of high-quality training data. We studied two applications where only noisy or low quality training data is available. In a first setting we presented the novel task of automated lyric annotation and an accompanying dataset providing explanations to lyrics and poetic text. These models generate explanations to held-out poetic text. Our newly introduced dataset is one of the largest of its kind and will stimulate research in text normalization, metaphor processing and paraphrase generation. In the second part of this chapter, we extended sequence-to-sequence models with the possibility to control the characteristics or style of the generated output, via attention that is generated a priori (before decoding) from a latent code vector. After training an initial attention-based seq2seq model, we used a variational auto-encoder conditioned on representations of input sequences and a latent code vector space to generate attention matrices. By sampling the code vector from specific regions of this latent space during decoding and imposing prior attention generated from it in the seq2seq model, output can be steered towards having certain attributes. This was demonstrated for the task of sentence simplification, where the latent code vector allows control over output length and lexical simplification, and enables fine-tuning to optimize for different evaluation metrics.

6.2 Future Directions

Although algorithms received much of the credit for ending the last AI winter, we believe new datasets and more efficient supervision will be essential for extending the present AI summer. Many exciting opportunities are still available for the perspective of weak supervision in combination with deep learning, transfer learning and data augmentation for NLP.

Information Extraction Distant supervision and the patterns used for our KBP experiments are just two sources of weak supervision. By accepting weak supervision under many different forms and handling noise using the appropriate methods behind the scenes, we can allow users to provide higher-level, more expressive input. We believe that end-to-end information extraction systems such as DeepDive and Snorkel are effective ways of enabling non-ML experts to quickly generate training data and enable use of deep learning models to new domains.

Keyphrase Extraction In Chapter 4 we showed the subjectivity of the concept of keyphrases and that annotator agreement is low compared to other NLP tasks. As noted in [1], the field would benefit from evaluation metrics less sensitive to variations of the keyphrases' surface form, which evaluate correctness on a more semantic level. A more suitable evaluation

for keyphrase extractors would be to let annotators compare sets of keyphrases generated by different models. A downside of this setting is that different models need to be evaluated separately, and optimizing towards this score is impractical. Our benchmark collections offer many opportunities to further investigate many of these issues.
Text-to-Text generation Applications of sequence-to-sequence models beyond machine translation are increasingly appearing. Success is closely tied to the availability of training data, this is why recently methods study training methods that do not require parallel text altogether [2, 3]. We expect many other applications of the paradigm to appear in the near future based on crowd sourced data or weak supervision.

Other promising directions for future research, which were less prevalent in our own research, for reduced supervision for NLP include transfer learning and zero shot learning.

Transfer Learning Transfer Learning has had a large impact on computer vision and has enabled many people to use pre-trained models for their own applications. Transfer learning has found its way to NLP. The choice of the pre-training task is very important as even fine-tuning a model on a related task might only provide limited success [4]. Other tasks, such as those explored in recent work on learning general-purpose sentence embeddings [5], might be complementary to language model pre-training or suitable for other target tasks. Pre-training will be most useful when the trained model can be applied to many target tasks.

Zero-shot learning Zero-shot, one-shot and few-shot learning, in which only a handful of training instances are used, are interesting upcoming research directions. Following the key insight from Vinyals et al. [6] that a few-shot learning model should be explicitly trained to perform few-shot learning, this task has seen several recent advances. However, a few-shot learning benchmark for NLP is currently still lacking which could evaluate how well existing few-shot learning models from CV perform for NLP.

We believe applications of reduced supervision in combination with deep learning models offer great potential, and are very excited about how weak supervision approaches will continue to be translated into more efficient, more flexible, and ultimately more usable systems for NLP and ML.

References

[1] K. S. Hasan and V. Ng. *Automatic keyphrase extraction: A survey of the state of the art*. Proceedings of the Association for Computational Linguistics (ACL), Baltimore, Maryland: Association for Computational Linguistics, 2014.

[2] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. *Style Transfer from Non-Parallel Text by Cross-Alignment*. arXiv preprint arXiv:1705.09655, 2017.

[3] G. Lample, L. Denoyer, and M. Ranzato. *Unsupervised Machine Translation Using Monolingual Corpora Only*. arXiv preprint arXiv:1711.00043, 2017.

[4] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. *How Transferable are Neural Networks in NLP Applications?* In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 479–489, 2016.

[5] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. *Supervised Learning of Universal Sentence Representations from Natural Language Inference Data*. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, 2017.

[6] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. *Matching networks for one shot learning*. In Advances in Neural Information Processing Systems, pages 3630–3638, 2016.

A

The Ghent University Slot Filling Systems

In this appendix we provide a more general description of the Ghent University Knowledge Base Population systems which participated in the TAC KBP shared task for two consecutive years. The KBP shared task is part of the NIST Text Analysis Conference and aims to evaluate different approaches for discovering facts about entities and expansion of knowledge bases. A selection of entities is distributed among participants for which missing facts need to be extracted from a given large collection of news articles and internet fora. Important components of these systems are query expansion, entity linking and relation extractors. We provide system descriptions of the 2014 system in section A.1 and the improved 2015 system in section A.2.

A.1 Ghent University-IBCN participation in TAC-KBP 2014 slot filling and cold start tasks

In our first participation at the Tex Analysis Conference (TAC) workshop several baselines for relation extraction were implemented in combinatino with of-the-shelf components for retrieval, linking and feature extraction. Next to linear sparse-feature classifiers, a first version of a CNN-based classifier was used for relation extraction. Our first KBP system obtained scores at the median of the ranked systems.

⋆ ⋆ ⋆

M. Feys, L. Sterckx, L. Mertens, J. Deleu, T. Demeester and C. Develder

Published in 7th Text Analysis Conference, Proceedings. p.1-10, Gaithersburg (MD), USA, 2014.

Abstract This paper presents the system of the UGENT_IBCN team for the TAC KBP 2014 slot filling and cold start (slot filling variant) tasks. This was the team's first participation in both tasks. The slot filling system uses distant supervision to generate training data combined with a noise reduction step, and two different types of classifiers. We show that the noise reduction step significantly improves precision, and propose an application of word embeddings for slot filling.

A.1.1 Introduction

This paper presents the system of the UGENT_IBCN team for the TAC KBP 2014 slot filling and cold start (slot filling variant) tasks. This was the team's first participation in both tasks. The slot filling system uses distant supervision to generate training data combined with a noise reduction step, and two different types of classifiers. We show that the noise reduction step significantly improves precision, and propose an application of word embeddings for slot filling.

Relation extraction is a vital step for information extraction. This task has received attention in TAC KBP for a number of years as the Slot Filling track, and as a vital sub-task of the recently-introduced Cold-Start track. As this is our first participation in the Knowledge Base Population - Slot Filling and Cold Start tracks, our system starts from previous work by other teams, in particular systems described in [1] and [2] which use facts from external databases to generate training data, also known as distant supervision.

Distant supervision has become an effective way for generating training data in the slot filling task, as proven in last year's top submission [3].

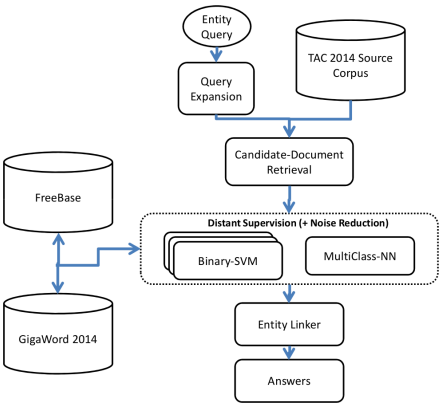


Figure A.1: 2014 KBP System Overview

The main contribution of our system is twofold: (1) we add a noise reduction step that boosts precision of the distant supervision step , and (2) we explore the use of word embeddings [4] for a relation detection classifier.

In the following Section, we give an overview of the system and describe different components. A more elaborate discussion of the training with distant supervision is given in Section A.2.3. Section A.1.5 discusses some adaptations of the system for the cold start task. Finally, results and a brief conclusion are given in Sections A.2.4 and A.2.5.

A.1.2 System Overview

Figure A.1 shows an overview of the slot filling system. Interactions between different components of the system and the different sources of data are visualized by arrows. In this Section we discuss those parts of the system which act at run time for the generation of slot fillers.

A.1.2.1 Query Expansion and Document Retrieval

The first step is the retrieval of all documents containing the entity query (person or organization) from the TAC 2014 source document collection. We expand the query by including all alternate names obtained from Freebase and Wiki-redirects for the given query. When we do not retrieve any alternate names, we clean the query, e.g., remove middle initials for persons and remove any company suffixes (LLC, Inc., Corp.) and repeat the search for alternate names using this filtered query. For indexing and search of the source collection we use Whoosh[1].

[1]http://pythonhosted.org/Whoosh/

A.1.2.2 Relation Classifiers

In each retrieved document we identify relevant sentences by checking if any of the entities from the expanded set of query entity names are present. Note that we did not use any co-reference resolution to increase recall, as suggested in earlier work [2, 5]. Next, we assign all slot candidates from the relevant sentences with a type (e.g., title, state-or-province). Slot candidates are extracted using the Stanford 7-class Named Entity Recognizer [6] and assigned a type using lists of known candidates for each type. For each combination of the extracted candidates with the query entity, we perform a classification of a type-matching relation from the TAC-schema. We trained four different sets of classifiers, i.e., two sets of binary Support Vector Machines (SVMs) and two multiclass classifiers, that resulted in four different runs submitted for the slot filling task. Only the first classifier was used for a run in the cold start task.

41 different binary SVMs are used to detect the presence or absence of each relation in the sentence for the query entity and a possible slot filler.

All binary SVMs trained for different relations use the same set of features, which is a combination of dependency tree features, token sequence features, entity features, semantic features and an order feature. We extract these features using Stanford CoreNLP tools [7]. This corresponds to the features used in [2]. A complete overview of the used features is given in Table A.1 using an illustration of the features for example relation-tuple <Ray Young, General Motors> and the sentence "Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi"[2].

The two sets of binary SVM classifiers differ in their training data. Classifiers for the second run use the original output from the distant supervision step, while the first set is trained on training examples after a noise reduction step. Section A.2.3 describes this training data in more detail.

A.1.2.3 Multiclass Convolutional Neural Network

We experiment with word embeddings and see if relation classification can benefit from their use in the classification task. Therefore, we implement a single Convolutional Neural Network (CNN) that functions as a multiclass classifier and we compare the performance of this network with the classification obtained by a logistic regression classifier with the same set of features, but without the use of word embeddings (discussed in the next Section).
The CNN only uses a subset of the features used by the SVMs, as shown in Table A.2. The network is trained on the cleaned training data, which we introduce later on.

We use the SENNA-embeddings [4] as word embeddings for the lookup tables. The length-varying WBO-features are modeled by a convolutional

[2]The same example sentence as used in [2]

Table A.1: Overview of different features used for classification for the sentence "Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi".

Feature	Description	Example Feature Value
Dependency tree	Shortest path connecting the two names in the dependency parsing tree coupled with entity types of the two names	PERSON←appos←officer → prep_of→ ORGANIZATION
	The head word for name one	said
	The head word for name two	officer
	Whether *1dh* is the same as *e2dh*	false
	The dependent word for name one	officer
	The dependent word for name two	nil
Token sequence features	The middle token sequence pattern	, the chief financial officer of
	Number of tokens between the two names	6
	First token in between	,
	Last token in between	of
	Other tokens in between	{the, chief, financial, officer}
	First token before the first name	nil
	Second token before the first name	nil
	First token after the second name	,
	Second token after the second name	said
Entity features	String of name one	Ray_Young
	String of name two	General_Motors
	Conjunction of *e1* and *e2*	Ray_Young–General_Motors
	Entity type of name one	PERSON
	Entity type of name two	ORGANIZATION
	Conjunction of *et1* and *et2*	PERSON–ORGANIZATION
Semantic feature	Title in between	True
Order feature	1 if name one comes before name two; 2 otherwise.	1
Parse Tree	POS-tags on the path connecting the two names	NNP→DT→JJ→JJ →NN→IN→NNP

Table A.2: The features used to train the multiclass classifiers.

Name	Description
wbo	words in between the two entities
bm	two words in front of the first entity
am	two words after the last entity
et1,et2	the entity types of the two entities
ntw	# words in WBO

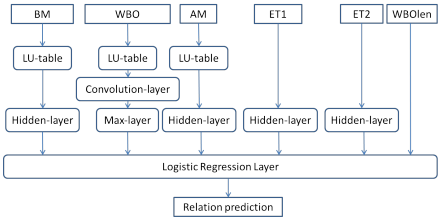


Figure A.2: Schematic representation of the CNN classifier.

layer combined with a max layer. A schematic representation of our network is shown in Figure A.2.

A.1.2.4 Multiclass Logistic Regression

The final classifier is a multi-class logistic regression classifier. It uses the same features as the CNN classifier (see Table A.2), and the same training data. The classifier is constructed to test the impact of the word embeddings on the classification results. This classifier consists of a single logistic regression layer and the features are a concatenation of the features given in Table A.2. The WBO-features are represented as a Bag of Words, thereby ignoring the order of the words. To reduce the number of dimensions, we do not use all word ids, but use only the 6.000 most representative words. The representative power was measured by the information gain of the words for the classification task, obtained with PearsonâĂŹs χ^2-test.

A.1.2.5 Entity Linking

Finally, the slot fillers extracted from the different documents are combined in an Entity Linking step. We link the entities from different documents and combine the extracted relation-tuples to obtain our final set of extracted relations. The output of this step consists of a list of all possible relation-tuples if the relation can have multiple tuples, e.g., for person_cities_of_residence. If only one relation instance is allowed, e.g., for city_of_birth, we choose the relation-tuple with the highest evidence. The

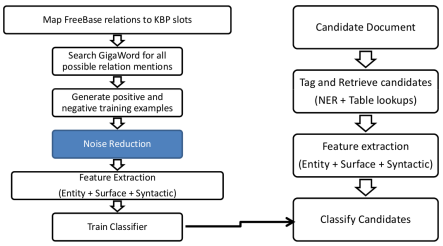


Figure A.3: Classifier Overview

evidence score for each relation-tuple is obtained by summing the evidence of all relation instances of this relation-tuple, i.e., the sum of the evidence-score given by the classifier for each sentence that expresses the relation-tuple.

A.1.3 Distant Supervision with Noise Reduction

Training data for the classifiers is generated using distant supervision. The left side of Figure A.5 shows the different steps for the generation of the training data. We start by mapping FreeBase relations to KBP slots and subsequently search the full GigaWord corpus for possible mentions of these relations, i.e., two entities from a fact tuple co-occurring in sentences. Negative examples are all co-occuring entities which are not present in FreeBase.

A.1.3.1 Noise Reduction

The training data obtained by distant supervision is noisy, i.e., not all extracted sentences that may indicate a given relationship actually express this relationship. e.g., , "President Obama visited Honolulu" does not express the relationship "per:city_of_birth", although president Obama was born in Honolulu. To address this problem, we add a noise reduction for 14 (due to time constraints) frequently occurring relationtypes. We let a team of students label approximately 2.000 distantly supervised instances per relation type with a true or false label. The fraction of instances labeled 'True' for the different relation types is shown in Table A.3.

The fraction of instances labeled 'True' strongly depends on the type of relation. To use these manually refined instances, we train a logistic regression (LR) classifier for each annotated relation and apply this classifier to the rest of the examples of the distant supervision output. For the relations with only a very small fraction of true examples (e.g., city_of_birth etc.) we did not train a LR classifier, but use a list of trigger words that need to be included, to filter the data. The results in Table A.4 show that this noise reduction step indeed results in a significant increase in precision.

Table A.3: Fraction of instances labeled 'True' for 14 relation-types

Relation	True fraction
per:title	91.78%
per:employee_or_member_of	90.90%
per:origin	86.48%
org:stateorprovince_of_headquarters	86.64%
org:top_members_employees	73.61%
per:age	65.60%
per:charges	58.97%
per:spouse	56.10%
per:countries_of_residence	58.97%
per:city_of_death	15.26%
org:founded_by	12.89%
per:cities_of_residence	2.15%
per:city_of_birth	1.29%
per:country_of_birth	0.30%

A.1.4 Subsampling

The dataset contains a lot more negative instances than positive instances. Therefore, we subsample the negative instances for each relation, to obtain approximately 50% positive instances and 50% negative instances for each relation.

A.1.5 Adaptations for the Cold Start Task

In our participation for the Cold Start Task (slot filling variant), our setup only required minimal modifications. We only used our SVM classifier with noise reduction for this task. The slot filling system and tagger were extended to handle relations involving Geopolitical Entities for the cold start variant. A first run is performed on the provided single-slot queries, and a second on slots for the resulting answers from the first run.

A.1.6 Results

A.1.6.1 Slot Filling task

The results for the slot filling task are shown in Table A.4. Our best results are the median score for this year's competition. We ranked at the 10th place of the 18 participating teams and ended as the 3th new team of the 6 new teams. Furthermore, by only evaluating based on the retrieved fillers (ignoring document id's) our F1 increased by 3.15%, resulting in a 9th place.

Table A.4: Results of the different runs on the slot filling task. b* stands for binary and m* for multiclass. NR represents the noise-reduction step.

	P	**R**	**F_1**
b* SVM + NR	24.1	15.9	19.2
b* SVM	16.4	16.3	16.3
m* CNN + NR	5.3	10.7	7.1
m* LR + NR	4.6	8.9	6.0
Median Score	25.8	16.1	19.8

Table A.5: Results of the different hops and the aggregate in the slot filling variant of the Cold Start task.

	P	**R**	**F_1**
Hop 0	24.7	16.6	19.9
Hop 1	7.5	4.9	5.9
All Hops	16.7	11.1	13.3

Binary SVMs clearly perform better than the multiclass CNN and the multiclass LR. This large difference between approaches is mostly due to the lack of extra features in the multiclass CNN. Unfortunately, this makes it difficult to assess the impact of using a multiclass classifier vs. the multiple binary classifiers. Note that the multiclass classifiers use the same filtered training instances of the binary SVM with noise reduction. The noise reduction is clearly beneficial for precision. By using the filtered training data, the precision of the binary SVMs improves from 16.4% to 24.1%, while recall only drops from 16.3% to 15.9%. The comparison of the CNN with the LR classifier shows a slight increase in F1 obtained by incorporating the word embeddings. However, both values are far below the F1 obtained with the binary SVMs. So far, we were unable to improve any results by using word embeddings. In future work we will test whether we can improve these results by also including additional features for the CNN network.

A.1.6.2 Cold Start task

The results for the different hops of the slot filling variant of the Cold Start task are shown in Table A.5. Precision and recall on the initial queries (Hop 0) are close to the performance on the regular slot filling task. In the second iteration (Hop 1) a lot of recall is lost, since a considerable fraction of these queries were not generated after the first run. Our system achieved second place out of three teams performing in the slot filling variant.

A.1.7 Conclusion

This paper described our first setup for the slot filling and cold start task which achieved results close to the median performance. We can conclude that the multiclass classifiers, only using lexical data, seriously underperformed the more elaborate binary SVMs and that we can significantly increase the performance of the classifiers by incorporating noise reduction of the training data obtained with distant supervision.

A.2 Ghent University-IBCN Participation in TAC-KBP 2015 Slot Filling and Cold Start Tasks

In our second participation we improved many of the components used in the system of the previous year. Especially in the relation extractors effort was put into generating clean training data, we described this procedure in more detail in Chapter 2. Depending on the evaluation metric, this system obtained the second, third or fourth highest scores out of 12 participating KBP systems. A highest scoring submission was based on the DeepDive system by the Database group Stanford University. Similar to our training data generation procedure, DeepDive generates training data based on feature annotations and regularizes by adding many weaker features to these instances.

⋆ ⋆ ⋆

L. Sterckx, T. Demeester, J. Deleu and C. Develder

Published in 8th Text Analysis Conference, Proceedings. p.1-10, Gaithersburg (MD), USA, 2015.

Abstract This paper describes the system of team UGENT_IBCN for the TAC KBP 2015 Cold Start (slot filling variant) task. The slot filling system uses Distant Supervision to generate training data for feature-based relation classifiers, combined with feature labeling and pattern based extractions. An overall performance 23.3% in micro-mean $F_1$was obtained, which is an increase of 10% compared to the team's 2014 participation.

A.2.1 Introduction

This was the second participation of team UGENT_IBCN in the Knowledge Base Population - Cold Start Slot Filling variant, the successor of the English Slot Filling track. Our system is based on the team's 2014 system [8] and uses techniques described in [9]. The relation extractor is based on Distant Supervision together with minimal amounts of supervision.

In the following Sections, we give a brief overview of the system and describe different components of the Knowledge Base Population system. A more elaborate discussion of the training with Distant Supervision is given in Section A.2.3. Finally, results and a conclusion are given in Sections A.2.4 and A.2.5.

A.2.2 System Overview

Figure A.4 shows an overview of the slot filling system. Interactions between different components of the system and the different sources of data

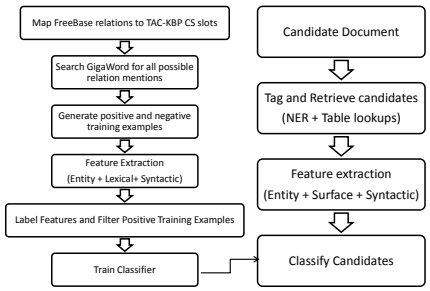


Figure A.4: 2015 KBP System Overview

are visualized using arrows. We discuss those parts of the system which act at run time for the generation of slot fillers.

A.2.2.1 Query Expansion and Document Retrieval

We first retrieve all documents containing entity queries (person or organization) from the TAC Cold Start 2015 source document collection. We expand the query by including all alternate names obtained from Freebase and Wiki-redirects for the given query. When we do not retrieve any alternate names, we clean the query, e.g., remove middle initials for persons, remove any company suffixes (LLC, Inc., Corp.) and repeat the search for alternate names using this filtered query. For indexing and search of the source collection we use the Whoosh[3] module for Python. This year no Named Entity Disambiguation was included, which resulted in wrong slot fillers for ambiguous entities, e.g., Gotham (New-York), Blues (Everton FC).

A.2.2.2 Named Entity Tagging

Each document was preprocessed using components of the Stanford CoreNLP toolkit [7]. In each retrieved document we identify relevant sentences by searching for any of the entities from the expanded set of query entity names. This year we include a co-reference module and resolve all synonymous noun phrases to a single entity. Noun phrases linked to any of the queries are used as subject entities for possible filler extractions. Next, we assign all slot candidates from the relevant sentences with a type (e.g., title, state-or-province). Slot candidates are extracted using the Stanford 7-class Named Entity Recognizer [6] and assigned a type using lists of known candidates for each type. Lists were expanded this year with those from the *RelationFactory* system [3].

[3]http://pythonhosted.org/Whoosh/

A.2.2.3 Relation Classifiers

For each combination of tagged entities with a query entity, we perform a classification of a type-matching relation from the TAC Cold Start schema. For classification we extract features from each candidate phrase and use binary Logistic Regression (LR) classifiers together with a small selection of High-Precision patterns.

Binary LR classifiers detect the presence or absence of a relation in the sentence for the query entity and a possible slot filler. All LR classifiers use the same set of features, which is a combination of dependency tree features, token sequence features, entity features, semantic features and an order feature. These correspond for the most part to the features used in [2]. A complete overview of the used features is given in Table A.1 using an illustration of the features for example relation-tuple <Ray Young, General Motors> and the sentence "Ray Young, the chief financial officer of General Motors, said GM could not bail out Delphi"[4].

Next to feature-based classification, a small selection of high precision patterns was used, some obtained from feature labeling and others from the *Relation Factory* KBP system [3]. If an exact match in the surface text between entities and a pattern is detected, the probability of the classifier is set to 1.

A.2.2.4 Entity Linking

In a final stage, the slot fillers extracted from the different documents are combined in an Entity Linking step. We link the entities from different documents and combine the extracted relation-tuples to obtain a final set of extracted relations. The output of this step consists of a list of all possible relation-tuples, if the relation is allowed to have multiple tuples, e.g., for person_cities_of_residence. If only one relation instance is allowed, e.g., for city_of_birth, we choose the relation-tuple with the highest probability assigned by the classifier. The evidence score for each relation-tuple is obtained by choosing the maximum evidence of all relation instances of this relation-tuple, i.e., the highest probability given by the classifier of all sentences that express the relation-tuple.

A.2.3 Distant Supervision with Feature Labeling

Distant supervision (DS) has become an effective way for generating training data in the slot filling task, as proven in many top-performing submissions in previous years [10]. In this year's competition we looked into ways of combining DS with minimal amounts of supervision.

The left side of Figure A.5 shows the different steps for the generation of training data. We start by mapping FreeBase relations to KBP slots and

[4]The same example sentence as used in [2]

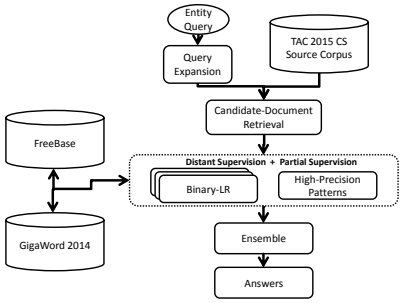


Figure A.5: Classifier Overview

Table A.6: Results on development sets.

	2013 ESF			2014 ESF		
	P	**R**	**F_1**	**P**	**R**	**F_1**
2014 Classifiers	42.8	19.7	27.0	28.0	18.6	22.4
2015 Classifiers	37.7	**37.2**	37.5	35.7	33.7	34.7
Patterns	**60.6**	12.1	20.2	**53.0**	8.7	14.9
Classifiers+Patterns	40.2	36.6	**38.6**	36.9	**35.9**	**36.4**

subsequently search the full GigaWord corpus for possible mentions of these relations, i.e., two entities from a fact tuple co-occurring in sentences. Negative examples are all phrases with co-occurring entities for relations which are not present in FreeBase.

Whereas in [8] instance labeling was used to self-train relation classifiers and reduce noisy mentions, we focus on learned features from an initial DS classifier. In a second stage, most confident positive features learned by the initial classifier are presented to an annotator with knowledge of the semantics of the relation and labeled as true positive, false positive (noise) or ambiguous. The collection of training instances is then filtered by only

Table A.7: Results of the different hops and the aggregate in the slot filling variant of the 2015 Cold Start task.

	Hop 0			Hop 1			All Hops		
Run	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
2014 - Best Run	24.7	16.6	19.9	7.5	4.9	5.9	16.7	11.1	13.3
2015 - Run 1 (High Precision)	**34.5**	25.0	**29.0**	**14.4**	9.9	11.7	**27.9**	19.8	**23.2**
2015 - Run 2 (Higher Recall)	33.0	25.2	28.6	12.5	10.6	11.5	25.4	20.2	22.15
2015 - Run 3 (Highest Recall)	28.0	**27.4**	27.8	13.1	**13.7**	**13.36**	22.7	**22.7**	22.7
2015 - Run 1 (Macro Mean)	-	-	34.29	-	-	13.3	-	-	27.0

including mentions with one of the true positive labeled features present, after which a second classifier is trained.

Our strategy is related to the *guidelines* strategy from Pershinaet al. [11], but instead of extracting guidelines using a fully annotated corpus, we label features entirely based on distant supervision. We then use a strategy from active learning literature, feature certainty [12] to rank and present features to the annotator, in order to further reduce the labeling effort. Feature Certainty is intuitively an attractive choice, as the goal is to reduce most influential sources of noise as quickly as possible e.g., for the relation *founded_by* there are many persons that founded the company which are also *top_members*, leading to many instances that we wish to remove when cleaning up the training data for the relation *founded_by*.

In the final set of classifiers an ensemble of two classifiers was chosen and confidences for relation extraction were averaged.

A.2.4 Results

A.2.4.1 System Development

The system was developed on data from the 2013 and 2014 English Slot Filling task. We found that important parameters to fine-tune, in order optimize F_1-scores, are classifier regularization, the ratio of true and false examples and the classification threshold. The highest micro-F_1scores obtained for these development sets are shown in Table A.6. Compared to classifiers used in 2014 participation in the English Slot Filling Task, large increases in performance (+10%) were attained.

A.2.4.2 Cold Start Results

Four runs were generated using the same set of classifiers. Submissions differ in the selection of thresholds set on the amount of fillers and confidence values. For each of the runs, at most, 10 fillers with the highest confidences were used to generate the second hop queries, this to reduce the generation second-hop fillers for wrong first-hop fillers. The micro-averaged P/R/F1 at each hop level for the different runs of the slot filling variant of the Cold Start task are shown in Table A.7. Compared to last year's participation an increase of almost 10% in $F_1$was obtained, placing fourth among 20 KBP systems from all variants and second out of twelve systems participating in the slot filling variant.

A.2.5 Conclusion

This paper described our second setup for the slot filling variant of the Cold Start task. We significantly increased the performance of our previous

relation extraction classifiers by incorporating noise reduction of the distantly supervised training data using feature labeling and high-precision patterns.

References

[1] M. Surdeanu, D. McClosky, J. Tibshirani, J. Bauer, A. X. Chang, V. I. Spitkovsky, and C. D. Manning. *A simple distant supervision approach for the TAC-KBP slot filling task*. In Proceedings of Text Analysis Conference 2010 Workshop. Citeseer, 2010.

[2] A. Sun, R. Grishman, W. Xu, and B. Min. *New York university 2011 system for KBP slot filling*. In Proceedings of the Text Analytics Conference, 2011.

[3] B. Roth, T. Barth, M. Wiegand, M. Singh, and D. Klakow. *Effective slot filling based on shallow distant supervision methods*. arXiv preprint arXiv:1401.1158, 2014.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. *Natural language processing (almost) from scratch*. The Journal of Machine Learning Research, 12:2493–2537, 2011.

[5] B. Min, X. Li, R. Grishman, and A. Sun. *New York university 2012 system for KBP slot filling*. 2012.

[6] J. R. Finkel, T. Grenager, and C. Manning. *Incorporating non-local information into information extraction systems by Gibbs sampling*. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 363–370. Association for Computational Linguistics, 2005.

[7] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. *The Stanford CoreNLP Natural Language Processing Toolkit*. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60, 2014. Available from: http://www.aclweb.org/anthology/P/P14/P14-5010.

[8] M. Feys, L. Sterckx, L. Mertens, J. Deleu, T. Demeester, and C. Develder. *Ghent University-IBCN participation in TAC-KBP 2014 slot filling and cold start tasks*. In 7th Text Analysis Conference, Proceedings, pages 1–10, 2014.

[9] L. Sterckx, T. Demeester, J. Deleu, and C. Develder. *Using Active Learning and Semantic Clustering for Noise Reduction in Distant Supervision*. In 4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014), pages 1–6, 2014.

[10] M. Surdeanu and H. Ji. *Overview of the english slot filling track at the tac2014 knowledge base population evaluation*. Proc. Text Analysis Conference (TAC2014), 2014.

[11] M. Pershina, B. Min, W. Xu, and R. Grishman. *Infusion of labeled data into distant supervision for relation extraction*. 2014.

[12] J. Attenberg, P. Melville, and F. Provost. *A unified approach to active dual supervision for labeling features and examples*. In In European conference on Machine learning and knowledge discovery in databases, pages 40–55, 2010.

B

Unsupervised Keyphrase Extraction

A disadvantage of supervised approaches is that they require training data and show bias towards the domain on which they are trained, undermining their ability to generalize well to new domains. Unsupervised approaches are a viable alternative in this regard. We make two focused contributions to the area of unsupervised keyphrase extraction by studying the use of topic models in graph-based word ranking models.

B.2 Topical Word Importance for Fast Keyphrase Extraction

In Section B.2 we improve on a state-of-the-art keyphrase extraction algorithm called topical pagerank (TPR). While the original algorithm requires a random walk for each topic in the topic model being used, ours is independent of the topic model, computing but a single pagerank for each text regardless of the amount of topics in the model. This increases the speed drastically and enables it for use on large collections of text using big topic models, while not altering performance of the original algorithm.

⋆ ⋆ ⋆

L. Sterckx, T. Demeester, J. Deleu, and C. Develder

Presented at the International World Wide Web Conference, Florence, Italy, 2015.

Abstract We propose an improvement on a state-of-the-art keyphrase extraction algorithm, Topical PageRank (TPR), incorporating topical information from topic models. While the original algorithm requires a random walk for each topic in the topic model being used, ours is independent of the topic model, computing but a single PageRank for each text regardless of the amount of topics in the model. This increases the speed drastically and enables it for use on large collections of text using vast topic models, while not altering performance of the original algorithm.

B.2.1 Introduction

Automatic Keyphrase Extraction (AKE) is the task of identifying a set of expressions or noun phrases which concisely represent the content of a given article. Keyphrases have proven useful for various Information Retrieval and Natural Language Processing tasks, such as summarization [1] and contextual advertising on web pages [2]. Currently two types of methods are used: supervised and unsupervised methods. State-of-the-art unsupervised methods transform the input document into a graph representation. Each node in this graph corresponds to a candidate-word and edges connect two candidates occurring within a certain text window. The significance of each node, i.e., word, is computed using a random walk algorithm based on PageRank [3]. The top ranked nodes are then selected to generate keyphrases. TextRank is one of the most well-known examples of a graph-based approach [4]. Recent work has shown that the quality of keyphrases is improved by using topic model information in the graph model. Topical PageRank (TPR) [5] is a variation on the TextRank-algorithm that incorporates topical information by increasing the weight of important topical

words based on the topic-document and word-topic distributions generated by a topic model. Experimental results showed that TPR outperforms other existing unsupervised AKE-methods. While TPR is an effective algorithm for the inclusion of topical information from the topic model, it requires a random walk for each topic in the topic model. This approach becomes cumbersome for huge collections of text using large topic models, as PageRank is a computationally intensive algorithm. In this paper we propose a modification of the original TPR algorithm which is equally effective but speeds up the algorithm as many times as the amount of topics in the topic model.

B.2.2 Single-Pagerank Topical Keyphrase Extraction

Topical PageRank, as described in [5], requires a PageRank for each topic separately and boosts the words with high relevance to the corresponding topic. In a word graph each candidate word (i.e., nouns and adjectives) become a vertex in set $v = \{w_1, \ldots, w_N\}$. For each candidate w_j, a window of the following words in the given article (typically chosen as 10) is selected and a directed edge from w_j to each word w_i included in the window is created, resulting in a directed graph. Formally, the topic-specific PageRank can be defined as follows:

$$R_z(w_i) = \lambda \cdot \sum_{j:w_j \to w_i} \left(\frac{e(w_j, w_i)}{O(w_j)} \cdot R_z(w_j) \right) + (1 - \lambda) \cdot P_z(w_i), \qquad \text{(B.1)}$$

where $R_z(w_i)$ is the PageRank score for word w_i in topic z, $e(w_j, w_i)$ is the weight of the edge $(w_j \to w_i)$, the number of outbound edges is $O(w_j) = \sum_{w'} e(w_j, w')$ and λ is a damping factor $\in [0, 1]$ indicating the probability of a random jump to another node. A large $R_z(w)$ indicates a word w that is a good candidate keyword in topic z. The topic specific preference value $P_z(w_i)$ for each word w_i is the probability of arriving at this node after a random jump, thus with the constraint $\sum_{w \in v} P_z(w) = 1$ given topic z. In TPR, the best performing value for $P_z(w_i)$ is reported as being the probability that word w_i occurs given topic z, denoted as $P(w_i|z)$. This indicates how much that topic z is focused on word w_i. With the probability of topic z for document d $P(z|d)$, the final ranking score of word w_i in document d is computed as the expected PageRank score over that topic distribution, for a topic model with K topics,

$$R(w_i) = \sum_{z=1}^{K} R_z(w_i) \cdot P(z|d). \qquad \text{(B.2)}$$

Adjectives and nouns are then merged into keyphrases and corresponding scores are summed and ranked. Note that original TPR requires a PageRank for each topic in the model. Since topic models with a large amount

of topics (e.g. $K = 1,000$) are reported to empirically perform best, this requires many computations for each document, especially for long ones. That is, for D documents the total amount of PageRanks for AKE is $K \times D$. We propose an alternative strategy to avoid this large computational cost, by using but a single PageRank per document. We do this by using a single weight-value we call $W(w_i)$ indicating the full topical importance of each word w_i in the PageRank instead of K topic-specific values and summing all results. First, we determine the cosine similarity between the vector of word-topic probabilities $\mathbf{P}(w_i|Z) = (P(w_i|z_1), \ldots, P(w_i|z_K))$ and the document-topic probabilities of the document, $\mathbf{P}(Z|d) = (P(z_1|d), \ldots, P(z_k|d))$, to determine the single weight value $W(w_i)$ per word w_i and document d.

$$W(w_i) = \frac{\mathbf{P}(w_i|Z) \cdot \mathbf{P}(Z|d)}{\|\mathbf{P}(w_i|Z)\| \cdot \|\mathbf{P}(Z|d)\|}. \tag{B.3}$$

This quantity $W(w_i)$ can be considered the 'topical word importance' of word w_i given document d, where the contribution of a particular topic z_k is larger if w_i is an important word for that topic, and the topic is strongly present in the considered document. As a result, the single PageRank $R(w_i)$ becomes

$$R(w_i) = \lambda \cdot \sum_{j:w_j \to w_i} \left(\frac{e(w_j, w_i)}{O(w_j)} \cdot R(w_j) \right) + (1 - \lambda) \cdot \frac{W(w_i)}{\sum_{w \in \nu} W(w)}. \tag{B.4}$$

B.2.3 Evaluation

To detect any change in performance, we use a dataset comprised of news articles built by Wan and Xiao [6], that contains 308 news articles from the 2001 Document Understanding Conference (DUC) summarization-track, with 2,488 manually assigned keyphrases. We create a mapping between the keyphrases in the gold standard and those in the system output using an exact match. We reduce keyphrases to their stems using the Porter-stemmer and use three standard evaluation metrics for AKE: precision, recall, and F1-measure. Other parameters (for the stemmer, tokenizer and PageRank) are identical to those in the original TPR-paper [5].

Figure B.1 shows how much our modification speeds up the computation time as compared to the original TPR algorithm for processing of the complete collection of articles. Both approaches are programmed using identical pre-processing functions and PageRank implementations. The graph shows the linear speed up achieved by making the algorithm independent of the amount of topics, and thus constant time. Figure B.2 shows precision-recall curves for the original TPR and ours using a single PageRank, using the same topic model of 1,000 topics trained on Wikipedia data (a corpus similar to the one used in the original TPR [5]), and two baselines TF-IDF and TextRank. The effectiveness of our method is close to identical

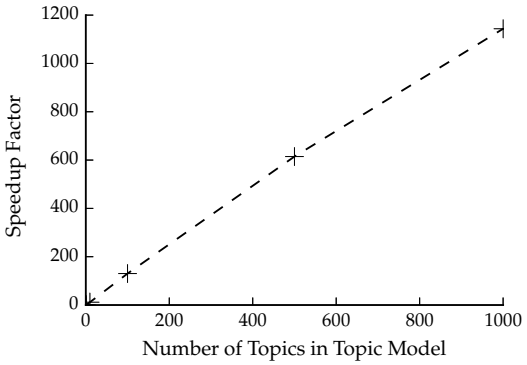
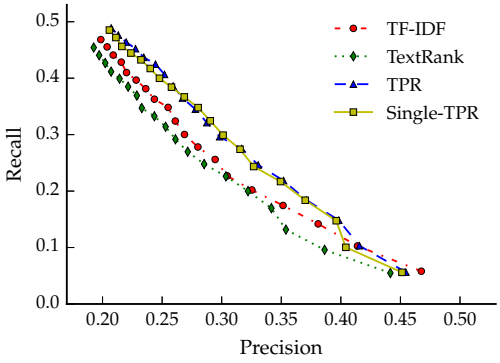


Figure B.1: Speed-up with proposed modification



.

Figure B.2: Comparison of the original TPR [5] (indicated 'TPR') with the more efficient single-PageRank TPR (indicated 'single-TPR'), and two baselines, TF-IDF and TextRank [4]

while computation time is reduced by factor $\approx 1/K$ (i.e., 1,000 times faster in this example).

B.2.4 Conclusion

We propose a more efficient use of topic models for unsupervised keyphrase extraction. Using a single value for topical word importance in a PageRank algorithm based on the cosine similarity between the vector of word-topic probabilities and the document-topic probabilities of the document, we achieve a constant computation time, independent of the topic model

being used. We show that this modification does not significantly alter the performance while reducing the computation time by a large margin.

B.2 When Topic Models Disagree: Keyphrase Extraction with Multiple Topic Models

In Section B.2 we explore how the unsupervised extraction of topic-related keywords benefits from combining multiple topic models. We show that averaging multiple topic models, inferred from different corpora, leads to more accurate keyphrases than when using a single topic model and other state-of-the-art techniques.

⋆ ⋆ ⋆

L. Sterckx, T. Demeester, J. Deleu and C. Develder

Presented at the International World Wide Web Conference, Florence, Italy, 2015.

Abstract We explore how the unsupervised extraction of topic-related keywords benefits from combining multiple topic models. We show that averaging multiple topic models, inferred from different corpora, leads to more accurate keyphrases than when using a single topic model and other state-of-the-art techniques. The experiments confirm the intuitive idea that a prerequisite for the significant benefit of combining multiple models is that the models should be sufficiently different, i.e., they should provide distinct contexts in terms of topical word importance.

B.2.1 Introduction

Keyphrases are defined as a set of terms or noun phrases which concisely summarize the content of a document. Automatic Keyphrase Extraction (AKE) has been beneficial for various applications such as document categorization and contextual advertising on Web pages. A distinction can be made between supervised and unsupervised methods. State-of-the-art unsupervised methods apply a graph-based approach. These methods build a graph from the input documents, each node corresponding to a candidate word and edges connecting two co-occurring candidates. Nodes or vertices are ranked according to their importance using a graph-based ranking method like PageRank. Top-ranked vertices are then combined to generate keyphrases. The inclusion of topical information has been shown to be beneficial for extracting keyphrases from documents. Liu et al. propose Topical PageRank (TPR) [5], a variation of PageRank that incorporates topical information by increasing the importance of highly relevant topical words based on Latent Dirichlet Allocation (LDA) [7]. Each word in the graph gets an additional weight (denoted as $W(w_i)$ for word w_i) in the random-walk algorithm proportional to the cosine distance of the topic-document distribution and word-topic distribution from the LDA topic

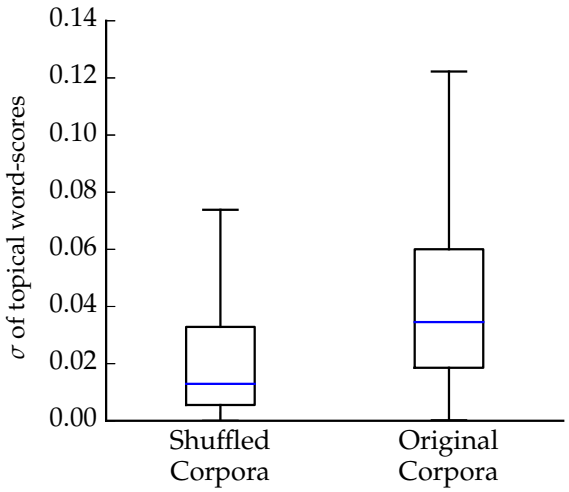


Figure B.3: Box plot displaying the average standard deviation for all topical word scores $\{W^c(w_i)\}_{c=1...4}$ for different topic models c, based on the original four collections ('Original Corpora'), versus four topic models based on a random equal share of all data together ('Shuffled Corpora')

model. Experimental results showed that TPR outperforms other unsupervised AKE-methods. We assess that topical importance strongly depends on the collection of training documents for LDA and their corresponding context. Specific words can be essential in one context yet only secondary in another. First we show that topical word importance varies with the corpus the topic model is trained on. Then we show that a simple combination of multiple different topic models and word scores leads to more accurate AKE results, a prerequisite being the diversity of the training corpora.

B.2.2 Disagreement by Topic Models

We demonstrate how we can improve the accuracy of a single-model TPR by combining information from multiple topic models. We use four different corpora to study the influence of the topic models on AKE: **Wikipedia** (a corpus similar to the one used in the original TPR contribution [5]), **Reuters Corpus Volume I (RCV1)** [8] (800,000 manually categorized newswire stories), **Wikinews** [1] (A free-content news source wiki, maintained through collaborative journalism, from February 2013) and **New-York Times** [9] (a collection of 300,000 NYT news articles). It is known that ensemble meth-

[1] http://en.wikinews.org/

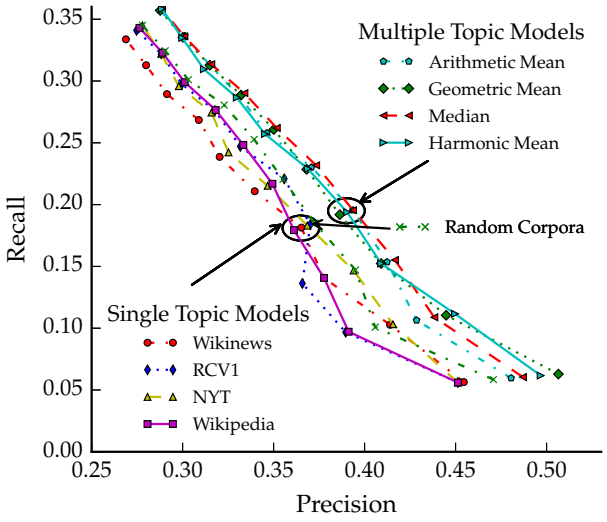


Figure B.4: Precision-recall curve for combinations versus single-model TPR for 1 to 10 extracted keyphrases.

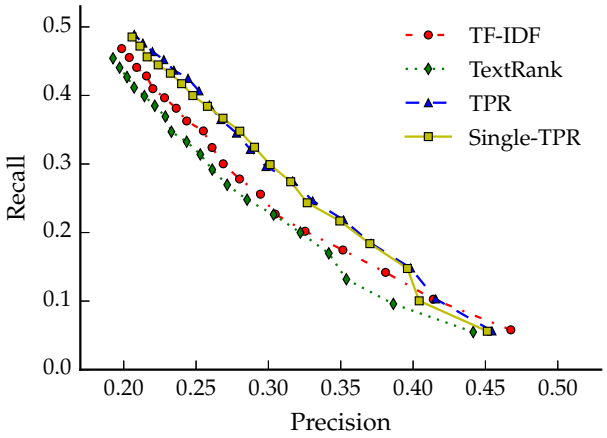


Figure B.5: MultiTM-TPR versus baselines for 20 extracted keyphrases

ods like model averaging obtain better accuracy than can be obtained from any of the constituent learning algorithms. We assess if and when this is the case for learning algorithms based on topic models for AKE. We first investigate how the topical importance scores from the word-document similarities, which are used in TPR, vary with the corpus the models are trained on. We then use this disagreement to make a combined weight applying several methods for averaging. Large test corpora for AKE, containing a broad set of topics, are hard to find and create. The creation of such a set is in progress, but we wish to report promising results on an existing, smaller set of news articles built by Wan and Xiao [6], that contains 308 news articles from the 2001 Document Understanding Conference (DUC) summarization-track, with 2,488 manually assigned keyphrases. The following experiment is conducted: next to training topic models on the original corpora, we reassign documents from each of the mentioned topic model corpora to one of four new collections randomly, and train a 1,000-topic LDA-model on all collections. As in [5], all of the models' vocabularies are reduced to 20,000 words. This results in four different topical word scores indicated as $W^c(w_i)$ with c denoting the index of the model being used. In Figure 1a, standard deviations of the four weights are shown for the shuffled and for the original corpora for each word in the 308 documents of the test-corpus. We observe that there is a much higher variance in the importance of the words between models when trained on the specific contexts of documents from the original collections. This means that different topic models trained on corpora with distinct contexts, used in TPR, will produce very different word scores and thus keyphrases, whereas topic models trained on more uniform contexts lead to similar keyphrase rankings.

B.2.3 Averaging Topical Importance

In the previous section the disagreement between models showed the dependence of topical word importance on the corpus the topic model was trained on. We now attempt to leverage this disagreement, composing word scores which reflect a more realistic importance of the words. For this purpose we apply several metrics which combine all weights into a single weight to be used in the PageRank for TPR. For this experiment, all models are trained on the full vocabulary of their respective corpora. We apply four ways of averaging the four weights: the arithmetic mean, the geometric mean, the harmonic mean and the median. We create a mapping between the keyphrases in the gold standard and those in the system output using an exact match. We reduce keyphrases to their stems using the Porter-stemmer and use three standard evaluation metrics for AKE: precision, recall, and F1-measure. Other parameters (for the stemmer, tokenizer and PageRank) are identical to those in [5]. The resulting averaged precision-recall curves for increasing numbers of assigned key-

phrases (ranging from 1 to 10 keyphrases) are shown in Figure 1b. The results of all single topic models are approximately equal. When averaging scores generated from topic models from these original corpora, a change in accuracy is noticed. For each separate combination between different topic models some increase was obtained. All ways of averaging reach a similar increase in performance with respect to the single models. When looking at the top keywords, a slightly higher precision is observed for those averaging methods that penalize values with more spread, like the harmonic and geometric mean. This increase in accuracy is not observed when randomizing the contexts of the different topic models as demonstrated in Section B.2.2, when there is less variance in the scores topical importance. A topic model was also trained on a single large corpus, consisting of all the single corpora, but this resulted in a similar performance obtained using one of the single topic models trained on a separate smaller corpus. We finally compare our new multi-topic-model method (denoted as 'MultiTM-TPR') to existing baseline methods in Figure 1c and the best single-model TPR. Our MultiTM-TPR outperforms baselines and the original TPR. Also for the highest scored keyphrases, where a single topic model TPR is inferior to the TF-IDF baseline. All improvements of MultiTM-TPR over other methods are verified, using bootstrap resampling, resulting in significance levels of $p < 0.05$.

B.2.4 Conclusion

In this paper we showed ongoing work demonstrating the benefit of combining multiple topic models for Automatic Keyphrase Extraction. We studied the influence of the corpus the topic model is trained on, and showed disagreement between models which are trained on different corpora. Averaging weights from several topic models leads to an increase in precision of extracted phrases. When training models, an important aspect is the difference in contexts between the corpora, which leads to different topic models and thus disagreement about word importance. We leverage this disagreement by computing a combined topical word importance value which, when used as weight in a Topical PageRank, improves accuracy of extracted keyphrases. Moreover, we show that this benefit of using multiple topic models is attained when the models differ substantially. For future work, we intend to research whether more sophisticated methods for combining or selection of specific models can be applied.

References

[1] E. D'Avanzo, B. Magnini, and A. Vallin. *Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004*. In Proceedings of the 2004 DUC, 2004.

[2] W.-t. Yih, J. Goodman, and V. R. Carvalho. *Finding Advertising Keywords on Web Pages*. In Proceedings of the 15th International Conference on World Wide Web, WWW '06, pages 213–222, New York, NY, USA, 2006. ACM.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank citation ranking: Bringing order to the web.* 1999.

[4] R. Mihalcea and P. Tarau. *TextRank: Bringing Order into Texts*. In Proceedings of the 2004 conference on EMNLP, 2004. Available from: http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf.

[5] Z. Liu, W. Huang, Y. Zheng, and M. Sun. *Automatic keyphrase extraction via topic decomposition*. In Proceedings of the 2010 Conference on EMNLP, pages 366–376, 2010.

[6] X. Wan and J. Xiao. *CollabRank: towards a collaborative approach to single-document keyphrase extraction*. Coling, pages 969–976, August 2008. Available from: http://dl.acm.org/citation.cfm?id=1599203.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. *Latent Dirichlet Allocation*. JMLR, 3(4-5):993–1022, 2003. doi:10.1162/jmlr.2003.3.4-5.993.

[8] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. *RCV1: A New Benchmark Collection for Text Categorization Research*. JMLR, 5:361–397, December 2004. Available from: http://dl.acm.org/citation.cfm?id=1005332.1005345.

[9] K. Bache and M. Lichman. *UCI Machine Learning Repository*, 2013. Available from: http://archive.ics.uci.edu/ml.