

July 2019

## Poetry: Identification, Entity Recognition, and Retrieval

John J. Foley IV

Follow this and additional works at: [https://scholarworks.umass.edu/dissertations\\_2](https://scholarworks.umass.edu/dissertations_2)



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

Foley, John J. IV, "Poetry: Identification, Entity Recognition, and Retrieval" (2019). *Doctoral Dissertations*. 1573.

[https://scholarworks.umass.edu/dissertations\\_2/1573](https://scholarworks.umass.edu/dissertations_2/1573)

This Open Access Dissertation is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

**POETRY: IDENTIFICATION, ENTITY RECOGNITION, AND  
RETRIEVAL**

A Dissertation Presented

by

JOHN FOLEY

Submitted to the Graduate School of the  
University of Massachusetts Amherst in partial fulfillment  
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2019

College of Information and Computer Sciences

© Copyright by John Foley 2019

All Rights Reserved

# POETRY: IDENTIFICATION, ENTITY RECOGNITION, AND RETRIEVAL

A Dissertation Presented

by

JOHN FOLEY

Approved as to style and content by:

---

James Allan, Chair

---

W. Bruce Croft, Member

---

Brendan O'Connor, Member

---

Joe Pater, Member

---

James Allan, Chair of the Faculty  
College of Information and Computer Sciences

## ACKNOWLEDGEMENTS

I would like to first thank my partner: Emma Tosch, without whom I would certainly have never completed (or started) my dissertation. This paragraph is woefully inadequate to express the depth of support and mentorship she has provided to me and others in our PhD journeys.

Next, I would like to thank my family: Mom, Dad, Mary Beth, April, Tuukka, Rocky, and Buddy. I would also like to thank my mother for teaching me my engineering sense: that instructions are for deciding whether you need leftover pieces after it's built, and my father for teaching me to never let my schooling interfere with my education.

I would like to thank my friends who are otherwise unmentioned for their inspiration and support: Adam; Alex; Ameer, Prashant, Neil & Sid; Ben; Bobby; Cibele; Dan C.; Eileen; Emma S.; Gwen; Kaleigh; Jeff; Laura; Luisa; Marc; Sam H.; Sam Burkart; Sam Baxter; and Zach.

Thanks as well to Simon Everhale, Carolyn Buck and Myung-ha Jang for their work that led to the first chapter of this thesis, and discovering that poetry was such a difficult genre. I would like to additionally thank Michael Zarozinski for numerous conversations about Proteus, Books, Galago and for helping label some poetry pages. Special thanks go to Hamed Zamani, who encouraged me to focus my entire thesis on poetry and stop trying to include other digital library work.

I would like to thank the two generations of CIIR students and alumnae who have become my friends: Ashish, CJ, Dan, David, Elif, Ethem, Hamed, Helia, Henry, Jae Hyun, Jiepu, Keping, Lakshmi, Liu, Marc, Martin, Michael, Mostafa, Myung-ha, Nada, Qingyao, Rab, Sam, Shahrzad, Sheikh, Shiri, Tamsin, Van, Weize, and Youngwoo. I would like to especially

thank Laura, Jeff, Sam, and Marc for being such a critical part of my research training. I apologize to the people and visitors I must have forgotten: it has been a joy to be part of such a large research community.

I would like to especially thank Jean Joyce for doing me countless favors. Her help has been irreplaceable, especially while I've been teaching at Smith. Dan Parker has kept the lights on and servers running while always providing pleasant conversation and sage advice. Thanks to Kate Moruzzi for always looking out for me and helping me get reimbursed. I would like to thank Joyce Mazeski as well for advice and access to the Chair's schedule. I would like to thank David Fisher for his confidence in me, and point out it will now be my turn to start every conversation with him using the phrase "Why are you still here?"

I would like to thank the Toybox Team (Emma Tosch, Kaleigh Clary & David Jensen) and the rest of KDL for a fun and successful distraction this past year.

Thanks to my colleagues at Smith College for being a source of support during my first year as a faculty member. Special thanks to R. Jordan Crouser, but also the entire CS department: Sahar Al-Seesi, Alicia Grubb, Katherine Kinnaird, Jamie Macbeth, Nicholas Howe, Joseph O'Rourke, Charles Staelin, Ileana Streinu, and Dominique Thiébaud.

I would like to thank my TAs for sharing in the Fall & Spring CSC212 journey with me: Artemis, Faith, Georgina, Grace, Hafsah, Jenny, Lauren, Logan, Meredith, Prayasha, and Tiffany. Special thanks goes to my Summer 2019 research assistants: Ananda & May for reminding me that poetry is cool. And I would like to thank all the Smith and earlier UMass students who ever expressed surprise that I was still a student as well. Your reactions were priceless sources of motivation.

I would like to conclude this chapter by thanking my committee. First, thanks to my advisor, James Allan. A long time ago he asked me what I would do with access to every book in the world; the belated answer to that question is this entire thesis. I would also like to thank the rest of my committee: Professors W Bruce Croft, Brendan O'Connor, and Joe Pater, who all provided useful, constructive feedback, as well as their excitement to this work.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11-PC20066, in part by NSF grant #IIS-1617408, and in part by a subcontract from Northeastern University supported by the Andrew W. Mellon Foundation. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## **ABSTRACT**

# **POETRY: IDENTIFICATION, ENTITY RECOGNITION, AND RETRIEVAL**

MAY 2019

JOHN FOLEY

B.S., UNIVERSITY OF MASSACHUSETTS LOWELL

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

Modern advances in natural language processing (NLP) and information retrieval (IR) provide for the ability to automatically analyze, categorize, process and search textual resources. However, generalizing these approaches remains an open problem: models that appear to understand certain types of data must be re-trained on other domains.

Often, models make assumptions about the length, structure, discourse model and vocabulary used by a particular corpus. Trained models can often become biased toward an original dataset, learning that – for example – all capitalized words are names of people or that short documents are more relevant than longer documents. As a result, small amounts of noise or shifts in style can cause models to fail on unseen data. The key to more robust models is to look at text analytics tasks on more challenging and diverse data.

Poetry is an ancient art form that is believed to pre-date writing and is still a key form of expression through text today. Some poetry forms (e.g., haiku and sonnets) have rigid



structure but still break our traditional expectations of text. Other poetry forms drop punctuation and other rules in favor of expression.

Our contributions include a set of novel, challenging datasets that extend traditional tasks: a text classification task for which content features perform poorly, a named entity recognition task that is inherently ambiguous, and a retrieval corpus over the largest public collection of poetry ever released.

We begin by looking at poetry identification - the task of finding poetry within existing textual collections, and devise an effective method of extracting poetry based on how it is usually formatted within digitally scanned books, since content models do not generalize well. Then we work on the content of poetry: we construct a dataset of around 6,000 tagged spans that identify the people, places, organizations and personified concepts within poetry. We show that cross-training with existing datasets based on news-corpora helps modern models to learn to recognize entities within poetry. Finally, we return to IR, and construct a dataset of queries and documents inspired by real-world data that expose some of the key challenges of searching through poetry. Our work is the first significant effort to use poetry in these three tasks and our datasets and models will provide strong baselines for new avenues of research on this challenging domain.

# TABLE OF CONTENTS

	Page
<b>ACKNOWLEDGEMENTS</b> .....	iv
<b>ABSTRACT</b> .....	vii
<b>LIST OF TABLES</b> .....	xiii
<b>LIST OF FIGURES</b> .....	xv
 <b>CHAPTER</b>	
<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 Outline & Contributions .....	4
<b>2. RELATED WORK</b> .....	<b>7</b>
2.1 Poetry Identification .....	7
2.1.1 Poetry Identification in Longer Works .....	7
2.1.2 Poetry in Document and Genre Classification .....	8
2.2 Named Entity Recognition in Poetry .....	10
2.2.1 Historical NLP .....	11
2.2.2 Domain-Specific NLP .....	12
2.3 Query Log Analysis .....	13
2.4 Poetry and Information Retrieval .....	14
2.4.1 A Brief History of Retrieval Models .....	14
2.4.2 Poetry Categorization .....	15
2.4.3 Music, Emotion & Sentiment .....	16
2.5 Discussion .....	17

<b>3. POETRY IDENTIFICATION</b> .....	<b>19</b>
3.1 Poetry Identification .....	20
3.2 Formatting and Content .....	21
3.2.1 Formatting Models of Poetry .....	22
3.2.2 Content Models of Poetry .....	23
3.2.2.1 Language Modeling .....	23
3.2.2.2 Neural Architecture .....	24
3.3 Experimental Setup .....	25
3.3.1 Modern Poetry Collection .....	26
3.3.2 In-Domain Data Collection .....	26
3.3.2.1 Source Book Dataset .....	26
3.3.2.2 Candidate Page Selection .....	27
3.3.2.3 Model Selection Dataset .....	29
3.3.2.4 Generalization Dataset .....	29
3.3.2.5 Page Level Label Distributions .....	29
3.3.2.6 Labeling Effort .....	30
3.4 Results .....	31
3.4.1 Poetry Identification Model Selection .....	31
3.4.2 Model Generalization Experiments .....	32
3.4.3 Discussion of Results .....	34
3.5 Poetry Dataset Curation .....	34
3.5.1 Model Efficiency .....	35
3.6 Discussion .....	36
<b>4. NAMED ENTITY RECOGNITION</b> .....	<b>37</b>
4.1 Background and Challenges .....	38
4.1.1 Background .....	38
4.1.1.1 News-Based NER on Poetry Data .....	39
4.1.1.2 Twitter-Based NER on Poetry Data .....	40
4.1.2 Challenges .....	41
4.1.2.1 No Sentences .....	41

4.1.2.2	Alternate Capitalization Patterns .....	41
4.1.2.3	Boilerplate text: HEADER .....	42
4.1.2.4	Non-traditional Entity Usage .....	43
4.2	A Dataset for Named Entity Recognition in Poetry .....	44
4.2.1	Page Candidate Selection .....	44
4.2.2	Poetry-NER Token Classes .....	44
4.2.3	Dataset Overview & Baseline Performance .....	45
4.2.4	Labeling Effort .....	46
4.3	Poetry NER Model .....	47
4.3.1	Sequence Prediction Model .....	47
4.3.2	Word Representations .....	49
4.3.2.1	Character-LSTM Representations .....	49
4.3.2.2	Handcrafted Word Features .....	50
4.3.2.3	Simple Attention .....	51
4.3.2.4	Word Generalization Layer .....	51
4.4	Experimental Setup .....	51
4.4.1	Stochasticity and Variance .....	52
4.4.2	Measure Selection .....	53
4.4.3	Multiclass Weighting and the ENT class .....	54
4.5	Results .....	54
4.5.1	Validity of ENT Summary Class .....	55
4.5.2	Label Quantity Study .....	56
4.5.3	POETRY, PROSE, & HEADER Detection .....	57
4.5.4	Feature Ablation Study .....	58
4.5.4.1	The Importance of CoNLL Data .....	61
4.6	Discussion .....	63
4.6.1	Remaining Challenges for Poetry NER .....	63
4.6.2	Implications for traditional NER .....	63
<b>5.</b>	<b>POETRY RETRIEVAL .....</b>	<b>65</b>
5.1	Analysis of Poetry Information Needs .....	66
5.1.1	Query Logs Analysis .....	66

5.1.1.1	Qualitative Results .....	66
5.1.2	PoetryFoundation Categories .....	70
5.2	Poetry Retrieval Models .....	74
5.2.1	Query Expansion Models .....	74
5.2.2	Semi-Supervised Expansion Model Combination .....	75
5.2.3	Emotion Vector Model .....	75
5.2.4	Categorical Vector Model.....	77
5.2.5	Result Pooling .....	78
5.3	Poetry 20 Query Dataset .....	79
5.3.1	Crowdsourced Label Collection.....	81
5.4	Retrieval Model Evaluation .....	85
5.4.1	Experimental Setup .....	85
5.4.2	Results .....	85
5.4.3	Emotion and Category Vector Performance .....	86
5.5	Remaining Challenges for Poetry Retrieval .....	89
5.5.1	Error Analysis and Deeper Results .....	90
5.5.1.1	Photography Results .....	90
5.5.1.2	Graduation Results.....	90
5.5.1.3	Doubt Results .....	90
5.6	Discussion .....	97
<b>6.</b>	<b>CONCLUSION .....</b>	<b>98</b>
6.1	Contributions .....	99
6.2	Future Work .....	100
<b>BIBLIOGRAPHY .....</b>	<b>103</b>	

## LIST OF TABLES

Table	Page
3.1 Handcrafted Features for Poetry Identification (§3.2.1). Scaled features refer to the count of the feature on the page in comparison to an average page of the book.....	23
3.2 Poetry Identification Dataset Overview. Random is a subset of the Random + Active Model Selection dataset. These were used in a 5-fold cross-validation setup to train and select initial models. The Generalization pages were fully held-out for all experiments.....	29
3.3 Distribution of labels collected while viewing pages that might be POETRY. This data gives a sense of the great diversity of content available in 50,000 books.....	30
3.4 Comparison of Poetry Identification Methods. Results are from 5-fold cross-validation of active-learning selected labels <i>or</i> from unbiased uniform Random sampling as a test set.....	32
3.5 Comparison of Poetry Identification Methods on held-out Generalization set of 951 pages that cover 500 additional books. Content based models struggle under this data collection method.....	33
4.1 NER Dataset Statistics. This table describes the parameters of our novel Poetry dataset in the context of a news corpus (CoNLL 2003) and a microblog corpus (W-NUT16). Our examples are at the page-level and are much longer than sentences or tweets. Our test corpus for CoNLL is “testa”. Although our dataset is small, it compares favorably with CoNLL in terms of unique terms, and W-NUT16 in terms of total tagged words.....	45
4.2 Performance of the Spacy multilingual model on our Poetry-NER dataset at the token level. Existing taggers are not better than random (AUC=0.5) on poetry data. Taggers are typically expected to be re-trained for new domains.....	46
4.3 List of Handcrafted NER Word Features.....	50

5.1	Most frequent queries including stemmed forms of {poetry,poem,poet} in raw, unstemmed format. ....	68
5.2	Distribution of labels across the most frequent 200 queries. ....	69
5.3	20 example queries classified into the broad categories of TOPIC, METADATA, EVENT, MOOD, and OTHER. Note that some queries have multiple labels; the frequency of each category is given in Table 5.2 .....	72
5.4	Frequency ordered categories present in poetryfoundation.org dataset, categories in the middle are elided only those with frequencies lower than 100 and higher than 1000 are present in this list.....	73
5.5	Statistics for the two retrieval corpora: poetryfoundation.org and our own extracted collection. On disk sizes are calculated with du -h, we did not actually search for duplicates in the poetryfoundation.org.....	79
5.6	Queries and Descriptions used for Poetry 20 Dataset .....	80
5.7	Agreement and Relevance information for our Poetry 20-Query Dataset .....	82
5.8	Performance, ordered by mAP, of various retrieval models on our Poetry dataset. ....	87
5.9	Per-Query Performance of Category and Emotion vectors, ordered by mAP of the Category vector approach.....	88

## LIST OF FIGURES

Figure	Page
1.1 A Poem printed in the middle of an essay by James Russell Lowell (Lowell, 1914). . . . .	3
3.1 A Poem printed in the middle of a Gardening Guide (Rockwell et al., 1917) . . . . .	20
3.2 Generic Neural Network Architecture. . . . .	24
4.1 A stanza of a poem printed on page 48 of “The Poet Scout” (Crawford, 1886). The <b>bolded spans</b> were identified as person entities, and the <i>italicized span</i> . “ <b>Keep</b> ” was identified as a miscellaneous entity. . . . .	39
4.2 A poem about Jonathan Swift in a book composing some of his memoirs and notes (Swift and Scott, 1824). . . . .	42
4.3 Graphical representation of our Poetry-NER model. Word representations are built from handcrafted features, an LSTM of character embeddings and word embeddings. This is generalized and sent through another LSTM before going through hidden layers to our multiclass and multilabel predictions. . . . .	48
4.4 Mean Entity, PER, LOC, ORG and MISC performance across 30 trials and trained for 60 epochs. . . . .	55
4.5 Mean, Median, Maximum and Minimum Entity performance across 30 trials and trained for 60 epochs. . . . .	56
4.6 Mean, Median, Maximum and Minimum Entity AUC over 30 trials trained for 60 epochs with varying sizes of training data. . . . .	57
4.7 Performance of HEADER, POETRY, PROSE and ENT classes. . . . .	58
4.8 Means of each feature setting over 30 trials trained for 60 epochs. . . . .	60
4.9 Ranking-based bar plot of all 30 trials of neural training for our feature explorations. . . . .	62



5.1	Alphabetized list of 136 <code>poetryfoundation.org</code> categories in our dataset. . . . .	71
5.2	A Kodak advertisement in a book on photography (Fraprie, 1915); this was identified as poetry by our algorithm, and is a false-positive search result for “photography”. This book contains 9 full pages of advertisements at the end of its content. . . . .	83
5.3	An author biography at the end of a book containing an interview with a local publishing company done by a University Library (Rather et al., 1994). This is another false positive identified by our algorithm, and comes up as a result for “graduation”. . . . .	84
5.4	A poem about the invention of photography that references Appelles, an ancient Greek painter. Note that if this poem were presented without the title, we would be unable to tell that it was describing poetry without significantly more effort. . . . .	91
5.5	A mention of the word “photography” that appears in the early ranks of our poetry search; this document does not contain poetry but mentions the topic. This is much more common for “photography” than other queries. . . . .	92
5.6	A mention of the word “photography” in a poem that is not actually about photography, but using it as a metaphor in the phrase “pen photography” to describe realistic fiction. . . . .	92
5.7	A poem (most likely a song) about a woman named “Kitty Casey” written in dialect about the emotional investment of a family in her graduation, amongst other possible themes. . . . .	93
5.8	The last stanza of a poem (continued from the previous page, which does not mention “graduation” about the “poignant glad-sad” experience of Graduating and leaving behind friends and experiences. . . . .	93
5.9	A poem about coming of age and being disappointed with ones’ achievements when the time for graduation has come after ignoring advice from ones’ elders. . . . .	94
5.10	A quote from a Shakespearean sonnet about “doubt” and “love”, which is highly ranked for both queries due to the length of the document, but at rank 1 for “doubt”. The book contains a quote for each day of the year. . . . .	95
5.11	A Christian poem about religious doubt and personal tribulations. Religion is a very common context for “doubt” in our collection of poetry. . . . .	95

5.12 A poem that is clearly influenced by religion and directly personifies many concepts, including “Doubt”. This poem is highly ranked in a unigram search for “doubt” but is not straightforward in interpretation. . . . . 96

# CHAPTER 1

## INTRODUCTION

Analyzing, organizing and searching information are the core tasks in natural language processing (NLP) and information retrieval (IR). In general, techniques developed in these fields make certain assumptions about the text being analyzed or searched: that it is clear and descriptive, that it is non-fiction, that it can first be broken into sentences and then parsed into logical structures or even that it contains legitimate, commonly-understood words. We propose to study poetry as a domain for IR and NLP techniques because poetry is capable of breaking all of these assumptions.

Some poems are written for mood or tone, where their goal is not transmitting any particular set of facts or information, but eliciting or evoking an emotion with words. Others have clearly-defined structure, (e.g., haiku and sonnets), but style is not mandatory; many poems eschew capitalization and form – they are essentially a list of words. Some poems are a mere handful of lines, others are the length of book. Some rhyme within modern languages, others only before translation or in now-unspoken dialects. Almost no poems are composed of sentences and phrases, which are often assumed and identified in the first phase of automated natural language processing tools.

Poetry is an interesting domain because it has similarities to informal and ungrammatical text like speech and social media posts, connections to political speeches and protest songs, and it is primarily about emotion and mood while potentially having both fiction and non-fiction elements in both long and short forms. As a different source of text, it is clear that models traditionally built on news or web data are going to struggle with the variety present in poetry and this makes it an interesting domain for study.

In Chapter 3, we present the challenge of identifying poetry from within longer works. This text classification task is challenging even to humans and we cannot achieve reasonable recall with typical content-based classifiers because poems contain a long-tail of subjects and terms – there will always be poems missed with such a method. We pursue the identification and extraction of poetry from a large collection of books. While some sources will state that they contain or possibly contain poetry in their metadata (e.g., title, headings) other poems are quoted without context or in the midst of another document (e.g., a poem quoted in a collection of essays by Lowell (1914), presented in Figure 1.1). We extract a large collection of over 800,000 poem instances from a set of 50,000 books. After de-duplication, we have 600,000 unique pages with poetry.

Poetry is used in order to discuss real world topics, sometimes through satire. In order to understand external references, we need to identify the entities mentioned in such works. Poetry also makes heavy use of simile, metaphor and allusion – potentially referencing other well-known works – in order to communicate emotion and intent.

In Chapter 4, we therefore look at the traditional NLP task of named entity recognition (NER), where the goal is to label the spans in text that refer to real people, places, organizations and things. Traditional approaches to NER are unsuitable for unstructured documents like poetry because almost all state-of-the-art approaches depend on sentence boundaries and capitalization for efficiency and understanding. Naturally, poetry, like “internet-speak” discourse, may not contain any punctuation or capitalization while still referring to real-world entities. We design a method to avoid classical preprocessing steps and to push punctuation and line breaks into the model itself, so that any available structure can be learned without being dependent upon it.

With identification and entity recognition in hand, we look at retrieval over poetry data in Chapter 5. In order to build useful poetry retrieval models, we study some sources of user data relating to poetry. Using the AOL and MSN query logs, we categorize the dimensions along which users typically search for poetry. We notice that poetry search is usually motivated

wishes to be told. Let us find strength and inspiration in the one, amusement and instruction in the other, and be honestly thankful for both.

The very earliest of Pope's productions give indications of that sense and discretion, as well as wit, which afterward so eminently distinguished him. The facility of expression is remarkable, and we find also that perfect balance of metre, which he afterward carried so far as to be wearisome. His pastorals were written in his sixteenth year, and their publication immediately brought him into notice. The following four verses from his first pastoral are quite characteristic in their antithetic balance: —

“ You that, too wise for pride, too good for power,  
Enjoy the glory to be great no more,  
And carrying with you all the world can boast,  
To all the world illustriously are lost!”

The sentiment is affected, and reminds one of that future period of Pope's Correspondence with his Friends, when Swift, his heart corroding with disappointed ambition at Dublin, Bolingbroke raising delusive turnips at his farm, and Pope pretending not to feel the lampoons which embittered his life, played together the solemn farce of affecting indifference to the world by which it would have agonized them to be forgotten, and wrote letters addressed to each other, but really intended for

**Figure 1.1:** A Poem printed in the middle of an essay by James Russell Lowell (Lowell, 1914).

by users wanting to identify a poem for a life event or holiday, such as the birth of a child, a graduation, or mother’s day. We then identify the need to search by metadata, by topic, and by mood. We build a dataset sourced from these queries and from categories created by humans in an online poetry collection.

We build a test collection of 20 queries and about 1300 relevance judgments and use it to explore the relative utility of utility of topical and emotional query models, focusing on query expansion techniques. We discover that poetry search is unlike other retrieval tasks, and the prior probability of documents that are likely to be relevant to someone is quite high, motivating future study of more personalized and specific information needs. Then we analyze the performance of different vector representations for retrieval, aiming at emotional words and a combination of emotional and other topical words. We find that these models struggle in comparison to powerful query expansion models.

## 1.1 Outline & Contributions

Our contributions are organized hierarchically, by chapter.

In chapter 2, we provide a discussion of work that is related to poetry identification, classification, entity recognition and information retrieval.

In chapter 3, we define the task of poetry identification from longer works. We select effective models and show that content based models do not generalize well. Leveraging our best formatting model, we then build the largest digital collection of poetry in the world.

***Contribution 3.1:*** *We introduce and develop a dataset of 2,814 pages covering 1,381 digitally scanned books labeled for the identification of poetry. This is the first freely-available benchmark for any poetry identification task.*

***Contribution 3.2:*** *We show that active-learning based label collection for poetry tagging leads to overconfidence and bias in results. We further show that by maintaining a proportion of labels collected by true random-sampling we are able to more accurately quantify recall of our identification approaches.*

***Contribution 3.3:*** *We construct a model for poetry identification based on handcrafted, formatting features which generalizes extremely well to novel data while also being efficient to train and execute.*

**Contribution 3.4:** *We develop a neural model for poetry identification that uses no handcrafted features, but demonstrate that this and all content-based models fail to generalize to unseen books.*

**Contribution 3.5:** *We create a collection of 600,000 pages with poetry using our strongest poetry identification tools from 50,000 books. Unlike most prior works classifying poetry, we make this full dataset available for future work in the public domain.*

In chapter 4, we explore named entity recognition (NER) on poetry. Motivated by the lack of capitalization and strict structure in poetry we explore a more structure-independent model that does not require sentence splitting or additional preprocessing steps. We evaluate the different features of a modern neural NER model on poetry data, and find that cross-training on existing NER datasets is the only critical feature.

**Contribution 4.1:** *We collect a novel NER dataset on poetry in order to create a new and challenging benchmark for NER. Our dataset covers 631 pages with 5,809 word-level tags.*

**Contribution 4.2:** *We provide a discussion on how to collect NER datasets in this domain, including the relative cost of labeling and how many labels are required for some learning effectiveness.*

**Contribution 4.3:** *We demonstrate that sentence splitting is not required for training effective NER models on traditional datasets, enabling us to skip many preprocessing steps while maintaining token-level effectiveness.*

**Contribution 4.4:** *We train an NER model that is capable of identifying poetry from prose (and boilerplate) at the token-level. Our model achieves a mean AUC of 0.946 on our test dataset, whereas off-the-shelf taggers perform approximately randomly.*

**Contribution 4.5:** *We empirically study the features necessary for an effective poetry-NER system. The most important need of modern NER algorithms is more data, and we find that, surprisingly, news-based NER data is most applicable to poetry and that noisier data from social media is less useful.*

In chapter 5, we turn to ad-hoc information retrieval as a task. We present the first query log study on user information needs in or about poetry on the AOL and MSN query logs. We identify and quantify types of searching behavior that guide our design of retrieval models.

We also study a set of tags from human curated poetry available on the internet. With these two real-world sources, we design an set of 20 queries (alongside 1,347 document judgments) to explore IR over our novel poetry collection. Unfortunately, it is prohibitively expensive to deeply explore recall in tasks like ours and future work should consider focusing on personalized recommendation and search tasks.

**Contribution 5.1:** *We present a query-log and category-based analysis that helps us to tackle problems in retrieval of poetry that are motivated by real human needs. We show that queries for poetry mostly break down into poetry desired for events, and poetry queries are typically refined by metadata, topic, mood and emotion.*

**Contribution 5.2:** *We develop a retrieval dataset over our poetry corpus aimed at ranking poems in response to a emotion and mood tags using crowdsourcing and pooling. Our dataset includes 1,347 document-based labels for 20 queries, which fully-judges 22 models to a depth of 10.*

**Contribution 5.3:** *We analyze the agreement and labeling task of designing a retrieval dataset on top of poetry data, identifying the challenge of having high prior probabilities of relevance with common queries. We determine that the relative relevance of poetry that is in-topic is difficult for annotators to assess and future work should explore alternative labeling schemes, such as pairwise preference in this domain.*

**Contribution 5.4:** *We evaluate a set of traditional query-expansion models on our novel poetry retrieval dataset. We find that expansion based on poetry data is most effective, but that generalized knowledge is also very useful for understanding topical queries.*

**Contribution 5.5:** *We evaluate and compare two vector-based approaches to encoding emotional and categorical information into a poetry retrieval model. Both approaches perform more poorly than typical query expansion approaches, and the categorical dataset is more effective. The emotion dataset may be too small for effective use in this broad domain, or it may be that more fine-grained emotional categories are needed.*

All of our datasets are publicly-available online<sup>1</sup>, code is available by request.

---

<sup>1</sup><https://ciir.cs.umass.edu/downloads/poetry/>



## **CHAPTER 2**

### **RELATED WORK**

Our related work section will parallel the structure of this thesis. First, we will discuss work most related to our poetry identification task (Chapter 3, §2.1), then we will discuss work related to named entity recognition for poetry (Chapter 4, §2.2), and finally we will discuss work related to our query log analysis and poetry retrieval task (Chapter 5, §2.3, §2.4). We provide a short discussion in Section 2.5.

Since we explore three different tasks in this dissertation we introduce some necessary background and related work here but also introduce more when tasks and approaches are most relevant.

#### **2.1 Poetry Identification**

Work relevant to poetry identification falls into two categories: work that performed some categorization of existing texts into poetry or not, and work that sought to extract sub-documents that were poetry which is our goal.

##### **2.1.1 Poetry Identification in Longer Works**

Underwood et al. (2013) present a study of genre in Hathi Trust books, and one of their genres is poetry, so their techniques could be used for determining if a given book is a book of poetry. However, they evaluate at the book level, so they are looking for books whose contents are mostly or entirely poetry. As we will discuss and demonstrated by the example in Figure 3.1, poetry identification at the page level is a different kind of challenge: although Underwood et al. labeled pages in sequence, it appears that the ultimate goal was correct

book-level labels. At the book level, there are many important textual clues that can be used, e.g., metadata. Basically, they are able to capture book-level metadata: it is unlikely that a book containing collections of poetry will not mention poems or poetry in the first few pages (which are usually the title, publishing information, and foreword information).

In a later interim report, Underwood (2014) presents a deeper analysis and page-level evaluate for genre detection, which includes poetry. Their predictions are publicly available<sup>1</sup>, but the corresponding text is not without collaboration with the Hathi-Trust to the best of our knowledge. In this work, they notice some of the same challenges as us with collecting training data from rare classes. They chose to tag whole volumes as training data and focused on optimizing for precision. We chose to tag a small number of pages from more books and focus on recall.

In a similar task, Lorang et al. (2015) use image classification approaches to try to extract poetry from scanned newspapers. More recently, Kilner and Fitch (2017) explore extracting poetry from scanned Australian newspapers, and base their features on earlier poetry recognition work (Tizhoosh et al., 2008). These works inspire features in our formatting-model of poetry (Chapter 3).

### **2.1.2 Poetry in Document and Genre Classification**

Recently, Chaudhuri et al. (2018) investigate separating latin prose and verse and find that specific stylistic structure is the key feature in classification. Their dataset is small and limited to specific classics already classified by hand. Jamal et al. (2012) presents a study of Malay poetry by theme and into poetry or non-poetry using support vector machines. Although they test a version of the poetry identification task, they do not do so from within longer works.

Singhi and Brown (2014) explore the differences and similarities between Wikipedia, news, lyrics, and poetry. They focus on the use of adjectives, and find that classifying into one of

---

<sup>1</sup><https://github.com/tedunderwood/genre>

{article, lyrics, poetry} to be quite challenging: achieving 0.67 accuracy for lyrics, and 0.57 for poetry while getting much better (0.80) for articles. These accuracies from adjective-focused language modeling demonstrate how difficult poetry identification can be. In error analysis, they find that some musicians, e.g., Bob Dylan, have much more poetic lyrics, and that they are more likely to be mis-classified as poetry. Since we focus on identifying and extracting poetry from text documents, there is little difference between lyrics of a song and poetry (when in print, one could argue for lyrics either being poetry or being something distinct) so we consider these tasks to be equivalent in order to limit the expertise required for judgments.

Choi et al. (2016) present an automatic subject-based tagging system based upon lyrics and user interpretations of lyrics. They found that user interpretations were more useful for subject classification than lyrics because lyrics are poetry, and are semantically ambiguous, and that user interpretations tend to be clearer and easier to analyze. They use 100 songs selected from 8 categories in order to explore balanced classification.

In general, all of these works focus on datasets that are both proprietary and small. Our poetry identification task, combined with retrieval directly offers an alternative to expensive manual collection techniques.

A similar line of work also includes genre identification, and although numerous works study genre on the web (Rosso, 2008; Chaker and Habib, 2007; Sharoff, 2010; Kumari et al., 2014) and in news domains (Petrenz, 2014), these works make the assumption that one document will have a singular genre. One work in genre identification considers scanned educational documents (similar to scanned books), and they train a line-based classifier to identify noisy text for the purpose of removing them to improve document clustering (Jang et al., 2017), which is similar to an approach we will take in Chapter 3 but for a very different task.

## 2.2 Named Entity Recognition in Poetry

Entities have played a key role in a number of challenges in the information retrieval community, including TREC tasks (De Vries et al., 2008; Balog et al., 2010; Demartini et al., 2010; Aslam et al., 2013) and TAC challenges (McNamee and Dang, 2009). Entity-aware ranking methods have been shown to achieve state-of-the-art results for improving ad-hoc retrieval (Dalton et al., 2014; Xiong and Callan, 2015; Dietz et al., 2017a).

Modern approaches to named entity recognition focus on word and character embeddings. Lample et al. (2016) showed that neural architectures without additional training data can match state-of-the-art results with handcrafted features and large gazetteers. Work in NLP moves quickly, and with most modern focus on different neural architectures and techniques such as attention (Vaswani et al., 2017), adversarial learning (Yang et al., 2018), multi-task learning for cross-lingual NER (Wang et al., 2017). Recently, extremely large models have captured the attention of researchers, but they are not practical for our needs (Devlin et al., 2018), due to their large memory requirements and so we focus on simpler, LSTM-based models.

There is an incredible amount of work on neural network models for NLP, as surveyed by Goldberg (2016). It is now dominated by LSTM approaches, of which the standard LSTM can be shown to outperform variants if properly trained (Greff et al., 2017). Realistically, speaking, there are only a handful of techniques for dealing with variable-length inputs in neural networks, and we will discuss a few different sequence adapters: recursive neural networks (or RNNs) (Rumelhart et al., 1986), bidirectional long short-term memory networks (or LSTMs) (Hochreiter and Schmidhuber, 1997), and simple addition (Mikolov et al., 2013) as alternatives for learning. Since this thesis does not propose novel architecture or sequence adaptation, we consider the internals of these units out of scope.

Interestingly, Won et al. (2018) recently found that for the identification of place names in challenging (historical) corpora, ensembles of NER tools and models performed much better than individual tools, perhaps as a form of regularization.

### 2.2.1 Historical NLP

A number of works explore the generalization of modern natural language processing tools to historical and literary collections. Bamman (2017) presents a table containing the results of many studies, which all indicate a significant 20-30% drop in performance. Historical sources tend to be quite small: the Tycho Brahe historical Portuguese corpus is one of the largest and contains 76 texts, with 3.3 million words (Galves and Faria, 2010), which is quite impressive given how expensive it is to collect linguistic annotations.

There have been some successes in generalizing part-of-speech tagging to historical content. Rayson et al. (2007) focus on improving a rule-based system on Shakespeare’s plays by tackling the unification of word variants. They see a 3% improvement on accuracy from automatically replacing words with modern spellings, and a 4% improvement by doing so manually, although they do not close the gap fully to in-domain data. Most follow-on works on historical english leverage a tool they published later, VARD (Baron and Rayson, 2008). Scheible et al. (2011) corroborate these kinds of improvements for early-modern German.

Yang and Eisenstein (2016) point out that word replacement fails to capture the changes in syntactic structure or word usage that might be confusing to algorithms trained on modern sources. They then propose and evaluate a feature-embedding approach (like the word-skipgram model) to predict features based on words in context that improves part-of-speech tagging. They also find that some robustness can be created by using word embeddings or brown clusters as features. Recently, neural NER approaches that start from word embeddings (Lample et al., 2016) have completely replaced the handcrafted features for which Yang and Eisenstein trained feature embeddings for better robustness. In this work, we present a neural model for NER on poetry but expect to compare to such approaches in the future as we explore more NLP tasks on poetry data.

Pennacchiotti and Zanzotto (2008) present study of historical Italian works using a part-of-speech tagger trained on modern newswire. 9 of the 14 historical documents they select are poetry rather than prose, and they find that from a lexical perspective, there’s no difference

in coverage in their dictionary – poetry and prose use roughly the same percentage of known words in this corpus. They evaluate part of speech tagging, but do not observe any trends on poetry vs. prose performance, seeing trends dominated by the age of the text, with earlier texts being harder for modern tools to analyze.

Since our poetry data is sampled from publicly-available digitally scanned books, most of it could be considered historical content. Therefore, our study of NER contributes to our understanding of how to generalize NLP tools to historical content, and our finding that a neural model can learn from both modern news-NLP and poetry from digitally scanned books suggests that a domain-independent story for NLP tasks may be improving.

### **2.2.2 Domain-Specific NLP**

Although it is well-known that NLP models struggle to transfer across domains and to dialects, most works still tend to focus on a single domain.

One of the domains in which natural language processing approaches explore less traditionally-structured text is on so-called microblog data, such as Twitter. Adapting part-of-speech labeling to this domain required 1800 labeled tweets and novel feature development (Gimpel et al., 2011). Work on named entity recognition required annotation of 2400 tweets, and Ritter et al. (2011) explicitly found that using out-of-domain training data lowered performance and they needed this data explicitly to train models that would rely less-heavily on capitalization. More modern methods on twitter NLP tasks have turned to neural approaches, e.g., sentiment analysis (Becker et al., 2017), NER (Lopez et al., 2017), event extraction (Farajidavar et al., 2017), etc.

Another domain in which work on NER and NLP tasks has received a lot of attention is the bio-medical domain (Leaman and Gonzalez, 2008; Krallinger et al., 2017). Following modern work in NLP, approaches in specific domains are also moving toward neural approaches (Habibi et al., 2017).

## 2.3 Query Log Analysis

Analyzing user behavior through query logs has a long history, dating back before modern web search (Bates et al., 1993). Some of the first web studies were done by Excite (Jansen et al., 1998) and AltaVista (Silverstein et al., 1999). Queries are often classified into broad categories: navigational, informational, resource or transactional (Broder, 2002), and different kinds of queries benefit from different features and retrieval models. This understanding of queries was backed up by the analysis of Rose and Levinson (2004), who had a hierarchy built around navigational, informational and resource information needs. These works provided fundamental understanding of how users imperfectly express their intent through queries in web search.

Most academic studies of query logs and user behaviors depend upon the releases of the AOL (Pass et al., 2006) and MSN (Craswell, 2009) query logs. Many users in the AOL log have been de-anonymized (Amitay and Broder, 2008), and as a result of this and the aging data, few modern studies refer to these logs. However, the utility of learning from user data means that studies continue to be performed on this kind of data, despite ethical considerations.

Systems that can automatically identify the intent of a users' query can help retrieve suitable results more effectively. This work is closest to our goal of understanding how users retrieve poetry. There are many works that aim to identify user intent, and this has been the subject of the "Query Representation and Understanding Workshop" at SIGIR (Li et al., 2011). Approaches include learning from click logs (Li et al., 2008), Wikipedia (Hu et al., 2009a), or other resources. Some works attempt to directly create a taxonomy of user intents (Yin and Shah, 2010) or to infer information from a query-flow graph (Bai et al., 2011). Since we aim for a qualitative understanding of user search behavior more than a model of their behavior (since the logs available to us are older), most modern work on query log analysis does not directly apply to our needs and goals.

We will use poetry words as a clue to collect queries that involve intent to find poetry, so as to analyze and understand user needs and goals with regards to this interesting domain.

## 2.4 Poetry and Information Retrieval

There is very little work on poetry-specific information retrieval. On the web, poetry retrieval is often satisfied by page metadata. Although a poem itself will not contain tags or the word “poem”, text available in the design of the page or in comments on the poem itself will enable users to find it. We focus on searching for poetry that may not have sufficient meta-text.

Although some studies use examples of queries that target poetry (Chen et al., 2015), example queries from a log (Bai et al., 2011), or as an example of content within books (Kazai and Doucet, 2008), poetry has not been the focus of much work in the retrieval community. The IR community has explored some strange and different applications, such as joke retrieval (Friedland and Allan, 2008) and similar chess position retrieval (Ganguly et al., 2014), but poetry is a relatively new venture outside of web search.

### 2.4.1 A Brief History of Retrieval Models

At *query time*, the goal is to take a users’ representation of an information need (usually a short text query, but sometimes longer queries, multimedia such as images, or documents) and to use it to rank the documents most likely to be relevant. Relevance can be difficult to define, so we do so through human annotation. The function used to assign scores to documents and rank them for presentation to a user is often called the retrieval model. The most commonly implemented retrieval model<sup>2</sup> is the BM25 model (Robertson and Walker, 1994; Robertson et al., 2009), which is centered around a weighting of terms based upon the frequency of their occurrence in documents and the collection as a whole. There are a

---

<sup>2</sup>Among open-source search systems, there are none without a BM25 implementation (Lin et al., 2016), and it is the default similarity in the Apache Lucene system, which is heavily deployed commercially.



number of other retrieval models of similar effectiveness and efficiency that are based on single terms (Ponte and Croft, 1998; Zhai and Lafferty, 2001; Amati and Van Rijsbergen, 2002).

The story of information retrieval since the invention of powerful unigram models has been a study of additional features: such as interactions between terms (Metzler and Croft, 2005), mixtures of different fields (Robertson et al., 2004), and query expansion, i.e. inferring additional terms that should be included in the query (Rocchio, 1971; Lavrenko and Croft, 2001). Another approach for improving ranking methods has been to incorporate external information as both text (Diaz and Metzler, 2006), and in more structured forms (Dalton et al., 2014; Xiong and Callan, 2015). Some features have survived the test of time and alternate domains, and others have not (Armstrong et al., 2009).

Recently, more and more work focuses on trying to learn functions for estimating relevance, sometimes called “learning to rank” (Burges et al., 2005; Liu et al., 2009). We leverage a tool for learning to rank called RankLib (Dang, 2015) in order to combine several retrieval models in a supervised fashion.

Some of the most recent advances come from neural networks for learning, where the goal is to learn useful ranking functions with little manual supervision (Mitra and Craswell, 2017; Zamani and Croft, 2016, 2017; Dehghani et al., 2017), however, these approaches require large amounts of training data and/or computation time, are mostly used in a re-ranking step, and are typically applied to the top  $k = 1000$  results of a BM25 ranker.

When we devise a ranking system for poetry (Ch. 5), we will present models derived from query expansion models (Rocchio, 1971; Lavrenko and Croft, 2001; Diaz and Metzler, 2006).

### **2.4.2 Poetry Categorization**

Although these works often refer to themselves as “Poetry Classification”, we refer to them as “Categorization” because given text known to be poetry, they attempt to assign category labels to the input poems. We do so to disambiguate from our classification task

of “identification” (§2.1.2), where given input text, we attempt to determine first if it is poetry or not. In these works, they usually already have the poetry corpus and are looking to further refine it into classes; we find these works more relevant to our retrieval models than our identification methods.

Poetry classification with machine learning techniques is a fairly common task that has been studied in a variety of languages e.g., Bangla (Rakshit et al., 2015), Arabic (Ahmed and Trausan-Matu, 2017) and Thai (Promrit and Waijanya, 2017). Features of poetry are also studied, e.g, meter in Persian poetry (Hamidi et al., 2009), authorship and time in Ottoman poetry (Can et al., 2011). Emotion is an important part of poetry and is represented in many classes, but some works study it explicitly, e.g., in Arabic poetry (Alsharif et al., 2013) and in Francisco de Quevedo’s work (Barros et al., 2013). The single-language or single-author private corpus of poetry is a typical experimental setup for these works, due to the high cost of gathering such a dataset. Our work should hopefully reduce the cost of creating these datasets and increase the number that are available publicly. A recent work on classifying Punjabi poems into four categories is not a survey, but does provide a table of recent work, language targeted, and features discussed (Kaur and Saini, 2017).

### **2.4.3 Music, Emotion & Sentiment**

Since we treat the lyrics in music as a form of poetry, some of the work done on music retrieval is relevant to this dissertation. Schedl et al. (2014) present a recent survey of directions in Music IR.

Guo et al. (2012) introduce a system for retrieving songs by melody and lyrics, in a “Query-by-singing/humming” or QBSH system. They first classify queries to select those that include lyrics or singing and use the finite number of songs in their database to improve a word recognizer, matching lyrics to songs as they decode words from the audio. This approach is therefore only suitable to poem queries that include an exact quote, a small portion of our analyzed query log.

A large amount of work has been done on the detection of music emotion, in both the audio and textual spaces. Recent surveys are available, but only parts of these works are relevant to poetry (Kim et al., 2010; Yang and Chen, 2012). Korst and Geleijnse (2006) present a method to extract lyrics for a particular song from the web by means of alignment across multiple pages containing lyrics, mostly to accomplish boilerplate removal. Hu et al. (2009b) present a study of mood classification based on lyric and audio features, determining that neither is optimal in all cases and a combination is often helpful. The mood classification they perform is similar to our ideas for a field-based retrieval model.

Emotion has also been explored on purely textual data, such as for blogs (Yang et al., 2007) and news corpora (Lin et al., 2007). While the widely-studied sentiment analysis (Liu, 2015) is a larger example of emotion classification, most works on sentiment focus on simple positive or negative assignments – as shown in a recent survey (Ravi and Ravi, 2015). Such works have limited applicability to our goal to retrieve “love poems” or “poems about ennui”. Recently, some more focus has been placed on understanding the emotions of the reader, rather than the author (Chang et al., 2016). Aktolga (2014) has a thesis on integrating emotional and temporal diversity into retrieval models, providing a roadmap for how to integrate some of the emotional diversity we might be able to extract from poetry into more traditional IR tasks.

We end up developing a lexicon-based approach to building emotion vectors from queries and documents based on the emotion lexicon created by Mohammad and Turney (2013) and made available by request through their website.

## 2.5 Discussion

Most existing work on poetry discussed in this chapter rely on closed datasets and are not directly comparable to our approaches. In some cases we have drawn inspiration from their approaches (Tizhoosh et al., 2008; Kilner and Fitch, 2017; Lorang et al., 2015), and in others we can only survey the related work. Our novel poetry datasets provide an opportunity for a

more scientific approach to some of these fields now that there will be a sizable public dataset available.

## CHAPTER 3

### POETRY IDENTIFICATION

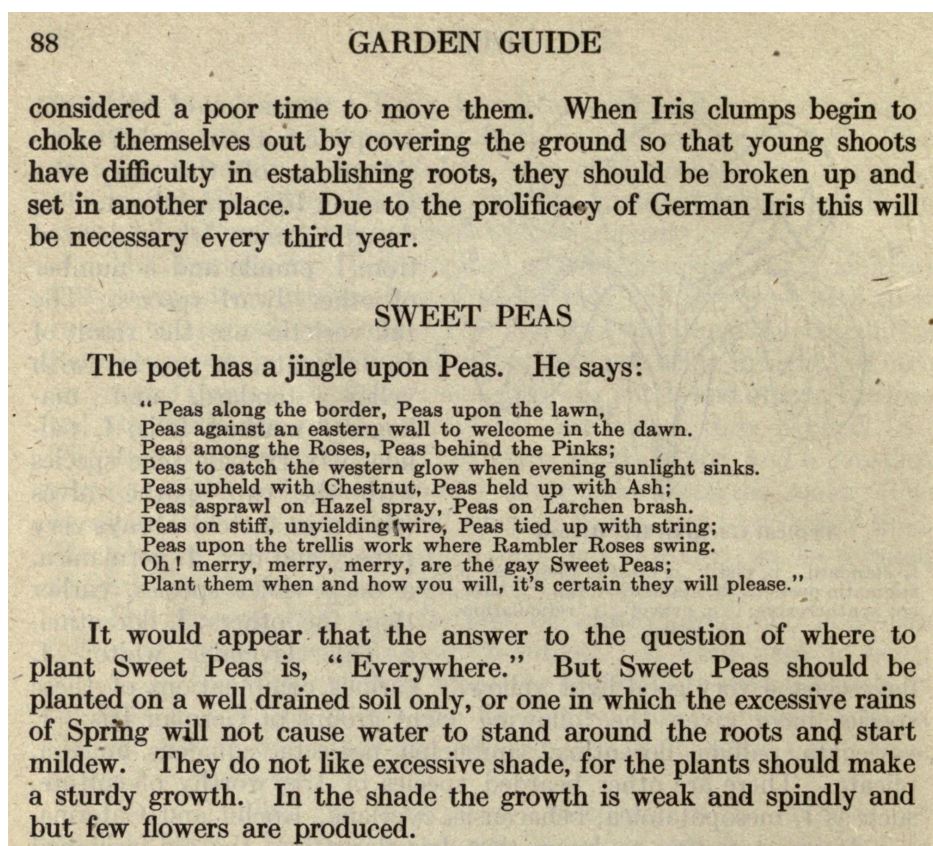
Although poetry is a well-known art form that is distinct from prose for humans, there is almost no work on how to automatically differentiate between prose and poetry. In this chapter, we explore and develop robust methods that operate independently on pages of books in order to determine whether these pages contain poetry. We first develop a model based on handcrafted, formatting features and then try to match this performance using content-based features, but find that content-based models generalize poorly.

Leveraging our best formatting-based models, we process 17 million pages in 50,000 books and find 800,000 unique pages of poetry. We then devise an efficient partial-alignment algorithm to unify duplicates and repeated publications of poems in order to have a more canonical dataset and to understand popularity in this large dataset. Although this dataset is relatively small by IR standards (2GB compressed with `bzip2`) - it represents the largest publicly-available digitized collection of poetry in the world. We discuss the collection and processing of this dataset further in Section 3.5. Given more CPU time we could generalize our approach to millions of books and hundreds of billions of pages.

In Section 3.1 we present an overview of our new task: poetry identification. In Section 3.2 we present our formatting and content models. In Section 3.3 we design our datasets and experiments and we present results in Section 3.4. We conclude in Section 3.6 with implications for future work on this dataset and collection fair datasets for more general text categorization tasks.

### 3.1 Poetry Identification

We define poetry identification as our task in this chapter because it allows us to find poetry that is effectively “hidden” within other works. In Figure 3.1, we demonstrate the difficulty we face: in the middle of a gardening guide from 1917 we discover a poem about “Sweet Peas”. No metadata for this book indicates the presence of poetry. It is unclear, even to a human, if “Sweet Peas” is the title of this poem or simply a section header within the book. Examples like this motivate and demonstrate the challenge of the poetry identification task.



**Figure 3.1:** A Poem printed in the middle of a Gardening Guide (Rockwell et al., 1917)

The challenge of poetry identification is relatively unstudied, as discussed in our related work (Ch. 2). Lorang et al. (2015) and Kilner and Fitch (2017) describe studies like ours, but they focus on a database of newspapers. Older works, including one that inspired some of our feature development (Tizhoosh et al., 2008), typically used sanitized datasets where some

documents were known to be poetry and others were known to be prose and classified them post-extraction. The canonical prior work on poetry identification is a comprehensive study of genre identification by Underwood (2014), based on a training set of 227 fully-annotated books; we focus on collecting fewer pages from an order of magnitude more books in order to try and maximize our poetry identification recall. We define poetry identification to be a task that unifies classification and extraction by doing our classification within larger works.

We call our task identification rather than classification to separate from a larger body of work on genre-identification for poetry (Jamal et al., 2012; Singhi and Brown, 2014; Choi et al., 2016) and related tasks like poetry generation whose models could potentially be used for classification but tend to focus on a specific genre of poetry (Díaz-Agudo et al., 2003; Yan et al., 2013; Veale, 2013; Zhang and Lapata, 2014; Ghazvininejad et al., 2016; Hopkins and Kiela, 2017; Yang et al., 2017).

## 3.2 Formatting and Content

Since our target domain is that of digitally scanned books, we know that it is likely that a human publisher laid out every page of every book. This means that when poetry is embedded inside another work, it was done with some kind of human intuition about how to format poetry, and how to set it off from the work that surrounds it, much as how the  $\text{\LaTeX}$  program (Lamport, 1994) formats our included image of poetry in Figure 3.1 with algorithms derived from human intuition and preferences about typesetting and page layout. We present our features and models in Section 3.2.1.

Unfortunately, focusing on formatting features ties us to the digital library domain. Poetry on the web, for instance, will be laid out in the tree-structure of HTML tags and there are far more options for formatting and style. The long tail of possibilities here make it difficult to envision style rules that will generalize. For adaptation to other domains containing poetry, we explore some content models of poetry in Section 3.2.2 but ultimately find that these generalize poorly (§3.4.2).

### 3.2.1 Formatting Models of Poetry

The core idea of this model is to detect anomalous formatting within a book. For instance, we look for pages in a book that have more or less punctuation than usual by designing our “scaled punctuation” feature. We also look at the various statistics, but especially the standard deviation for counts based on capitalized words, lines, and left and right margins. The idea of these formatting features is to identify pages that may have poetry or another included sub-document. Despite the lack of content features, these formatting features are able to find poetry quite well in a robust manner.

We design formatting features in four larger categories. As these feature categories are quite specific to books, we again note that formatting features are not going to generalize well to other domains and therefore inspect content-based models in the next section (§3.2.2).

**Capitalization features** capture the intuition that quite often every line of a poem or stanza will be capitalized, which may be more than regular text.

**Margin features** capture the intuition that poetry will be indented. That is, we expect poetry laid out in books to have a larger left margin (typically) in comparison to prose or introductory text.

**Page features** capture the intuition that lists, tables, and indexes may look like poetry in terms of margins and capitalization, but are likely to use punctuation and digits (numbers) more frequently. This section of features was designed to limit false positives.

**Book features** capture the intuition that pages within a book may be very different from their neighbors or very similar, and either of these may be indicative. Location in the book, as a fraction of the total length, helps identify book content from metadata, as content pages tend to be in the middle of a book.

We present a deeper survey of our formatting features in Table 3.1. Features that are marked as “Stats” include six statistical summaries of the set of values: the mean, minimum,



maximum, total-count, sum, and standard deviation of the feature. Our formatting model attempts to learn common intuitions and expectations of poetry formatting and how they deviate on the same page and across the book with other pages.

Group	Feature	Count
Capitalization	Fraction of Characters Capitalized	1
	Capitalized Lines Stats	6
	Capitalized Words Stats	6
Margins	Leftmost Term Start Stats	6
	Rightmost Term End Stats	6
	Words Per Line Stats	6
Page	Character Digit Fraction	1
	Character Punctuation Fraction	1
	Number of Words	1
Book	Scaled Punctuation	1
	Scaled Length	1
	Location in Book	1
Total	–	37

**Table 3.1:** Handcrafted Features for Poetry Identification (§3.2.1). Scaled features refer to the count of the feature on the page in comparison to an average page of the book.

### 3.2.2 Content Models of Poetry

We explore two content models of poetry: language modeling based on unigrams and a neural architecture approach. We collect some modern poetry data from the web (the `poetryfoundation.org` set) and prose data from Project Gutenberg books in order to have a larger textual training set in order to feed these more data-hungry methods, particularly our neural approach (§3.2.2.2).

#### 3.2.2.1 Language Modeling

The most obvious content model for any classification task is a language model. Learning weights on terms from a positive class and a negative class has worked well for a variety of tasks in information retrieval, including spam detection, and is the basis of many retrieval models (Croft et al., 2010). In prior work, we observed good performance from such approaches

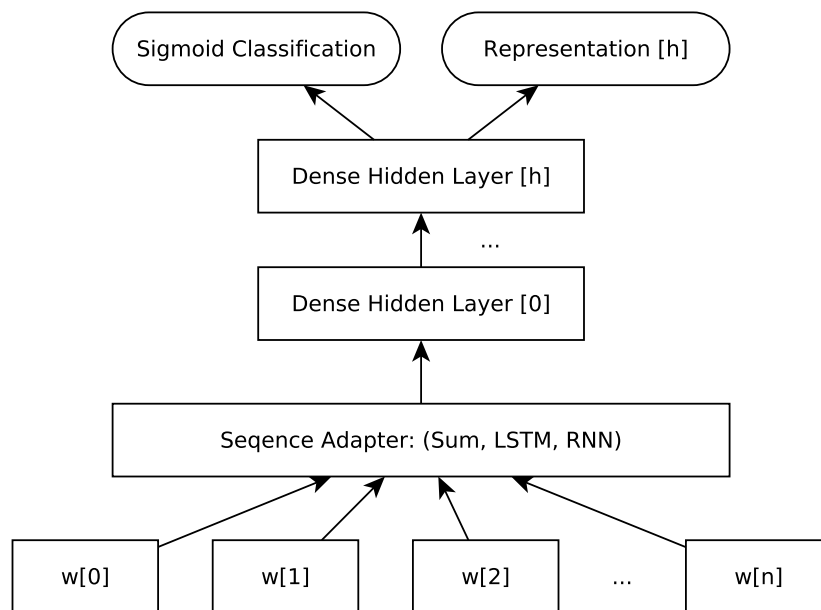
to modeling times and events within digitally scanned book data (Foley and Allan, 2015), and so this is a logical baseline for our task. We present some of the mathematics behind language modeling classifiers in Section 5.2.4, when we use it as the basis for topical vectors.

### 3.2.2.2 Neural Architecture

Modern approaches to text classification revolve around neural networks, where words are represented by unsupervised or supervised embedding vectors and the network learns to combine these word vectors into a document representation that is then used for classification, often with dot products or cosine similarity.

We use the largest dataset of poetry we could find in order to train semi-supervised representations for each line of poetry and use a stacked classification strategy to learn to predict the presence of poetry for the whole page based on its component lines.

A representation of our neural network is presented in Figure 3.2.



**Figure 3.2:** Generic Neural Network Architecture.

At the bottom of our network, we embed either words or character trigrams of the input. One of our motivations for exploring character trigrams is that we hoped that word shape

would be more informative and easier to learn from a smaller vocabulary. In both cases, we add special begin and end tokens to the input so that the network can identify patterns that start or end lines.

Above the level of embeddings we have a *Sequence Adapter*. There are a number of techniques for dealing with variable-length inputs in neural networks, and we explore a few different sequence adapters: recursive neural networks (or RNNs) (Rumelhart et al., 1986), bidirectional long short-term memory networks (or LSTMs) (Hochreiter and Schmidhuber, 1997), and simple addition (Mikolov et al., 2013) as alternatives for our learning.

We explore RNNs as a *sequence adapter*, specifically because they have been shown to be effective in poetry generation tasks (Zhang and Lapata, 2014; Ghazvininejad et al., 2016; Hopkins and Kiela, 2017; Yang et al., 2017).

After adapting our sequence to a fixed-size representation, we stack a number of hidden layers (chosen as a hyper-parameter) above this output. We use 50% dropout to help with regularization, as is standard for neural network training (Srivastava et al., 2014).

Our final layer depends on our objective. For pre-training with our unlabeled poetry, we use a single neuron with sigmoid output to predict whether a sample was drawn from a prose dataset or our poetry dataset. When we actually train and test our final classifier, we find that the single output neuron did not translate across domains consistently (sometimes it was quite good and other times its AUC barely achieved random), so we leverage the complete representation available at the last hidden layer in order to stabilize our performance.

### 3.3 Experimental Setup

In this section we discuss the variety of datasets we use for poetry identification pre-training, model selection, and generalization testing, as well as the rigor used in collecting our in-domain data. Results on these datasets will be presented in Section 3.4.

### 3.3.1 Modern Poetry Collection

Although there are many academic works that study poetry and poetry classification in a wide variety of languages, we were unable to find any that make their dataset public. Given the techniques and code we present, these researchers should be able to create an effective classifier from their existing collection and lower the cost of creating and open-sourcing these kinds of datasets.

There is a publicly available crawl of <http://poetryfoundation.org> along with code available on Github (Bridges, 2015). We decided to use this pre-existing crawl rather than to crawl the website ourselves, as it appears the API used may no longer be available. This dataset contains 506,209 non-blank lines of poetry in about 12,959 poems, which makes it relatively small in terms of IR resources.

These poems contain topical classifications, titles, authors, and poems segmented into lines in a computer-readable JSON format. Unfortunately, these poems have no visual, formatting, or layout information.

For an equally-sized collection of clean book data, we took the 150 most popular books from Project Gutenberg whose metadata did not contain either “poem” or “poetry”. We make our selection of books available alongside our code for this thesis to enable future work. We note that this constructed task is quite simple, so all of our neural approaches quickly learn to differentiate between poetry and prose, although the utility of their representations vary, and it is those representations that we truly compare in this section.

### 3.3.2 In-Domain Data Collection

In this section, we discuss how we collected poetry identification task data from a large collection of digitally scanned books.

#### 3.3.2.1 Source Book Dataset

We use a selection of 50,000 digitally scanned books selected randomly for the INEX Book Search task (Kazai and Doucet, 2008) for which we had previously aligned the documents to

their still publicly-available internet archive versions (Foley and Allan, 2015). This dataset contains 17 million pages, and 0.8 million of those pages have poetry according to our best classifier.

### 3.3.2.2 Candidate Page Selection

If you randomly select a page from a library of books, it is quite unlikely you will find a poem on this page. Because this prior probability of encountering our positive class is so low, and machine learning techniques perform best with many examples of positive data, we needed to devise a strategy for sampling that would maximize our learning ability while minimizing cost and bias. Unlike, e.g., e-discovery (Grossman et al., 2017) where many thousands of labels may be collected, the digital library domain tends to have a much more limited budget. Therefore, we knew we must borrow the principles of approaches such as of Continuous Active-Learning (Cormack and Grossman, 2016) where we mix in randomness with prediction from a classifier, without being able to quite achieve their theoretic guarantees on total recall.

In short, using a search engine to collect books with the words “poetry” in them is not scientifically sound. While you are likely to collect more positives faster, you are also biasing your collection toward “easier” data points: collections or archives of poetry, rather than “embedded” examples, like in Figure 3.1. We did this in a preliminary dataset (not used in this thesis) for developing our handcrafted features, although we decided to construct a more fair and robust training and evaluation set. The caution we developed during this early study served us well, and our interesting results on our generalization dataset (to be described) validate the care we take to collect balanced data.

We collected data using a mix of active learning, book-based heuristics, and random sampling techniques. The following list describes our methods for selecting candidate pages to be labeled.

1. **Metadata matching.** We randomly selected pages from books whose metadata contained terms that stemmed to “poem” or “poetry”.
2. **Page matching.** We also collected pages that mentioned the terms “poem” or “poetry”. Regardless of the ranking algorithm used, these are samples from the candidate set of a naïve full-text query – our best proxy for page-level metadata.
3. **Query Expansion.** As positive labels were collected, we constructed a Relevance model (Lavrenko and Croft, 2001). We then constructed an “expanded query” with no original query by sampling  $k \in \{5, 10, 100\}$  feedback terms from this probability distribution.
4. **Random page selection.** We chose a page with uniform random probability from all 17 million in our collection (excepting those which had been previously labeled). While the other methods in our strategy are all focused on improving our probability of encountering positive labels, we included a method that would simply suggest random pages at each point. While purely labeling random pages was extremely inefficient, maintaining a subset of the data that was truly collected randomly means that we can actually estimate the prior probability of poetry and tell whether our performance on random data is better or worse than data collected through a different method.

Selecting randomly from these four strategies allowed us to increase the balance of labels collected without committing to a particular active learning methodology. Although these methods introduce their own biases (e.g., the content-based *Query Expansion* method tended to recommend pages from the same book, once it learned the name of a protagonist in an epic poem) we knew which labels came from which methods and we could therefore control for this. When we later construct train, validation, and test splits we do so at the *book* level and have a separate setup that focuses on random selections as a test set.

Dataset	Pages	Poetry Pages	Books	Usage
Random + Active §3.3.2.3	1818	535	884	Model Selection
Random §3.3.2.3	352	29	359	Bias Estimation
Generalization §3.3.2.4	954	341	500	Only Testing

**Table 3.2:** Poetry Identification Dataset Overview. Random is a subset of the Random + Active Model Selection dataset. These were used in a 5-fold cross-validation setup to train and select initial models. The Generalization pages were fully held-out for all experiments.

### 3.3.2.3 Model Selection Dataset

For our training dataset, we collected 1818 page labels, of which 535 are poetry. Only 14 positives were found through pages which contained the word poetry, and 29 were found from random pages (of 359 total random samples). Pages with poetry as book metadata were 130 positive labels and the rest (366) were found with our relevance modeling approach. We used this data to evaluate and select our best set of models (§3.4.1). This dataset corresponds to the first two rows of Table 3.2.

### 3.3.2.4 Generalization Dataset

We later collected another 951 page labels in 500 new books, sampling from both uncertain (confidence scores of 0.49 – 0.51) and certain (confidence scores of  $\geq 0.90$ ) outputs from our two strongest models. We used this dataset in order to measure the generalization power of our best models (§3.4.2) – while retraining with this additional data would be helpful for improving our models, it serves as a truly-held-out test set and was built to be challenging. This dataset corresponds to the final row of Table 3.2

### 3.3.2.5 Page Level Label Distributions

In order to properly evaluate our task, we constructed a web-based interface that displayed a JPEG image of the scanned page to the user and asked them to assign a genre label to the document, starting with POETRY or PROSE. A variety of genres were encountered, including those considered poetry: “POETIC-PLAY”, “LYRICS”, and “POETRY” as well as others that would potentially be confusing: “INDEX”, “TABLE-OF-CONTENTS” and

“RECIPE” pages. Our user interface allowed annotators to add labels of their own design, and to skip challenging examples.

Frequency	Label
135	<i>various categories &lt; 10</i>
12	SKIP
12	TABULAR
13	EDUCATION
21	PLAY
23	LYRICS
26	ADVERTISEMENT
28	NOT-FOOD-RECIPE
35	TABLE
38	POETIC-PLAY
57	IMAGE
60	INTERVIEW
77	TABLE-OF-CONTENTS
78	BLANK
80	INDEX
178	PROSE
288	RECIPE
801	NOT-POETRY
857	POETRY

**Table 3.3:** Distribution of labels collected while viewing pages that might be POETRY. This data gives a sense of the great diversity of content available in 50,000 books.

Due to the mix of active learning approaches and query-based approaches, we were able to collect a fairly balanced dataset of poetry and not-poetry content, as well as some near-duplicates, such as “RECIPE” pages, which are visually somewhat similar, but topically very different<sup>1</sup>.

### 3.3.2.6 Labeling Effort

Collecting poetry identification labels was not a particularly difficult task. Since we timestamped label collection we can estimate the time spent on each document by comparing

---

<sup>1</sup>In a preliminary study, our classifier had a hard time with “RECIPE” pages and so we used our active learning approaches to increase the quantity of those found. Removing this extra data had no effect on evaluation.



timestamps of labels collected consecutively by the same annotators (including the latency required to load the page image). We ignored differences of greater than five minutes. The median time spent on a page label was 12 seconds and the mean time was 18.9 seconds. The 95th percentile time was 54 seconds and the 5th percentile time was 5 seconds. This indicates that in-situ poetry identification is quite simple for humans, at least on average.

## 3.4 Results

In this section, we discuss the results of our two experimental phases. In Section 3.4.1 we will discuss our model selection, where we used random sampling and simple active learning to construct a dataset (§3.3.2.3). In Section 3.4.2 we will present results from our generalization testing, where we explicitly collected data from 500 new books to test the generalization power of the best models (§3.3.2.4). We then discuss our thoughts about the results in Section 3.6 before moving on to constructing a large dataset with our best classifier in Section 3.5.

### 3.4.1 Poetry Identification Model Selection

Model evaluation results on our initial dataset (§3.3.2.3), collected with both active learning approaches and random selection are presented in Table 3.4.

When we train and evaluate different kinds of models on our poetry identification task, we find a wide range of performance from content-based approaches. Although a traditional language modeling approach to text classification does not perform well (barely better than random) we found compelling performance with the LSTM-based models, particularly the Character-Trigram LSTM models, which appeared to beat the formatting model on a limited dataset. A hybrid model including both kinds of features seemed to perform best.

For the formatting content features, we confirmed that non-linear modeling approaches were needed to capture our identification model, but did not see significant differences between random forest (RF) models and a neural, multi-layer perceptron (MLP) model with a small

Approach	Method	Random + Active		Random	
		AUC	$F_1$	AUC	$F_1$
Content	Language Modeling	0.546	N/A	0.607	N/A
	Word-Sum	0.876	0.681	0.550	0.156
	Word-RNN	0.907	0.800	0.673	0.353
	Word-LSTM	0.946	0.863	0.856	0.476
	Char-Sum	0.842	0.616	0.720	0.198
	Char-RNN	0.881	0.781	0.672	0.271
	Char-LSTM	0.955	0.897	0.910	0.724
Formatting	Random Forest (RF)	0.943	0.911	0.889	0.708
	MLP(16,8)	0.941	0.801	0.923	0.708
	Linear-SGD	0.881	0.655	0.495	0.167
Hybrid	RF + Word-LSTM	0.963	0.928	0.930	0.727
	RF + Char-LSTM	0.973	0.901	0.950	0.793

**Table 3.4:** Comparison of Poetry Identification Methods. Results are from 5-fold cross-validation of active-learning selected labels *or* from unbiased uniform Random sampling as a test set.

number of hidden nodes except that the random-forest had slightly lower performance on the random pages.

### 3.4.2 Model Generalization Experiments

Our generalization experiments take us beyond the set of randomly-collected pages (§3.3.2.3) to a wider set of unseen books (§3.3.2.4). Results are presented in Table 3.5, with the random dataset from Table 3.4 on the right and the new, held-out generalization set on the left.

Although our model selection experiments favored content-based approaches, our generalization data tells a different story. When we delved deeper into the results and collected pages from 500 new books, we found that the content-based approaches that had been doing so well could not generalize – in fact, they do barely better than random, suggesting that even though we strove to separate books into training and test datasets and looked at the subset of labels collected that were truly random, these models overfit to the topics they were exposed to in training.

Approach	Method	Generalization		Random (Table 3.4)	
		AUC	$F_1$	AUC	$F_1$
Content	Word-Sum	0.688	0.598	0.550	0.156
	Word-RNN	0.526	0.540	0.673	0.353
	Word-LSTM	0.558	0.516	0.856	0.476
	Char-Sum	0.587	0.521	0.720	0.198
	Char-RNN	0.519	0.451	0.672	0.271
	Char-LSTM	0.572	0.467	0.910	0.724
Formatting	Random Forest (RF)	0.941	0.823	0.889	0.708
	MLP(16,8)	0.902	0.612	0.923	0.708
	Linear-SGD	0.846	0.768	0.495	0.167
Hybrid	RF + Word-LSTM	0.923	0.803	0.930	0.727
	RF + Char-LSTM	0.922	0.803	0.950	0.793

**Table 3.5:** Comparison of Poetry Identification Methods on held-out Generalization set of 951 pages that cover 500 additional books. Content based models struggle under this data collection method.

We can see that of the formatting models, the Random Forest model is the most robust to the overfitting that plagued other models, especially our content-based models. Interestingly, the Linear model is much stronger on this dataset (perhaps reflecting that it could not learn from the folds in the smaller, “Random” set). Although the hybrid models of formatting and content still perform well, it is clear from individual performances that the formatting model is carrying all of the predictive power.

While we hoped that the word-LSTM and Char-LSTM models were capturing intuitions about how language within poetry works, these results suggest that they will need a lot more training data in order to capture something about the content of poetry that will generalize to unseen books. The model that holds up the best is the Word-Sum model, which is essentially a bag-of-words model based on word embeddings, but it is still far from good performance. The results of this experiment convinced us to collect our larger poetry dataset using our random forest model (Section 3.5).

### 3.4.3 Discussion of Results

One reason that the word and character models could be doing so poorly on new data is that they are facing an overwhelming quantity of previously unknown terms - so called out-of-vocabulary or OOV words. While this is a reasonable hypothesis for the word-based models, it does not hold for our char-LSTM models (one of the reasons it may be performing slightly better in terms of AUC), however, the embeddings learned for characters are clearly too specific.

For the word models, we explored using pre-trained word embeddings, like the GloVe 4B token set and holding them fixed. This forces the model to learn a transformation from GloVe space to poetry identification and would hopefully generalize better. Unfortunately, performance with pre-trained word embeddings could never get above 0.8 AUC regardless of settings, putting it significantly below our simpler and less expensive formatting models. It is likely that the issue is not one of pre-trained embeddings but that all pre-trained embeddings are loosely based on the point-wise mutual information of term pairs within traditional prose. If in the future we use our larger dataset to construct word embeddings, this strategy may prove more helpful.

Our retrieval experiments in Chapter 5 will suggest that some form of topical classification may be helpful for improving our formatting features, since certain queries had a much higher incidence of pages that did not truly contain poetry (see Table 5.7).

## 3.5 Poetry Dataset Curation

Given our formatting model, we now have the ability to automatically process large quantities of scanned digital books in order to curate a large collection of poetry.

We first create a collection of poetry from the 50,000 internet archive books that are most frequently used in literature (Kazai and Doucet, 2008; Foley and Allan, 2015). This processing completed on a single machine in under an hour and led to 800,000 pages identified as poetry.

This suggests that the true prior probability of poetry on a random page is going to be close to  $P(\textit{poetry}) = \frac{800,000}{17,000,000} = 4.7\%$ . On the other hand, roughly 2 out of every 5 books we processed contained some pages identified as poetry.

Upon beginning retrieval experiments in this dataset (Chapter 5) we found that poetry popularity obeys a typical Zipfian distribution and that we needed to perform some kind of duplicate detection and processing in order to provide diverse search results to users.

We developed a duplicate-detection method based on term hashes and longest-common subsequence which was partially inspired by existing work on duplicate detection in digital libraries (Yalniz et al., 2011). Despite spending some time optimizing our code, our duplicate detection work took about one day on a cluster of 50 machines, since there were 6.4 billion pairs. We found that there were 600,000 unique poems in our dataset after this processing. We use the 600,000 unique poem dataset for our retrieval experiments in Chapter 5, due to the expense of running duplicate detection jobs  $O(n^2)$  on a larger dataset of poetry. We have begun to process larger collections and plan to release them in the future.

### 3.5.1 Model Efficiency

Our formatting features (Table 3.1) and model are quite efficient, even though some features are scaled to the whole book, most features are computed independently per-page. We are able to apply a random forest algorithm and extract features for hundreds of books in just a few minutes. On a cluster with 200 machines, we were able to process a 100 million pages in just over 3 hours with 568 parallel jobs. The processing is mostly bottle-necked by the scanned book’s XML format and the cluster’s I/O.

On the other hand, our content models based on neural networks required significant memory and CPU footprint. Processing the 3000 pages in our labeled set took several minutes on a CPU machine – this was 50x slower than the formatting-based model. Although this performance is improved using GPU machines, it does not improve by a factor of 50x and these models still generalize poorly. In the future, we hope to use the labels from the

formatting model to train a much deeper content-based model, and only fine-tune this model using our human-label set.

Recent deep language models from NLP, such as the BERT model are trained on a much larger quantity of data and have shown greater performance on some tasks. These models have hundreds of millions of parameters and are meant to be fine-tuned for a particular model with smaller data. Unfortunately these models are still not very practical as they require a large amount of GPU memory<sup>2</sup> for fine tuning and prediction. We leave this exploration to future work after such models are further optimized and understood.

### 3.6 Discussion

One of the key challenges of poetry identification is topicality, and the likelihood of content classifiers to overfit. Despite following best practices with cross-fold validation and testing sets, we found apparently excellent results that did not generalize, in part due to the need to use active-learning to collect our dataset of poetry pages.

Our findings serve to highlight the importance of principled data collection and the need for large, robust datasets to be truly held-out from training and model design. Specifically, we find excellent results with content based approaches via neural networks that do not generalize. In addition, we find that deploying these models to a large collections is both inefficient and ineffective.

We hope the challenge of our identification dataset attracts interest from researchers interested in these modern approaches, text classification techniques, and in collection design.

---

<sup>2</sup>BERT requires over 12GB of GPU RAM (Devlin, 2018)

## CHAPTER 4

### NAMED ENTITY RECOGNITION

To a new language learner, achieving understanding of literature can be a *Sisyphean* – or impossible task. This is because of the use of cultural references or *allusions*, e.g., to the underworld punishment of the Greek king Sisyphus who was forced to roll a boulder up a hill only for it to roll down again. Language – especially poetry – is full of these cultural references.

External references are also important for understanding of current events (consider how many events are titled in reference to the Watergate hotel) or important figures (poems and songs often satirize political figures). This means that general text understanding is going to require the understanding of informally-structured text and the references it contains.

In order to understand discourse, we must be able to identify references to external content – and to understand the emotional connotations of these references. Our first step to allusion recognition and understanding is to explore named entity recognition (NER). NER is a common task in natural language processing (NLP) that aims to identify *entities* or the people, places, organizations and other things referenced in the text. While there are certainly other types of allusions that could be made in poetry, we consider this as the first step toward deeper reasoning about poetry.

In this chapter, we explore named entity recognition on poetry. On prose, neural approaches to this task dominate state-of-the-art approaches, and it has been shown that competitive performance can be achieved with no feature engineering at all (Lample et al., 2016). We explore a similar neural approach to NER but include some handcrafted features to determine if this approach is suitable for poetry as well as prose.

In Section 4.1 we discuss general approaches and background needed to understand named entity recognition approaches. We also introduce unique challenges of poetry. In Section 4.2 we discuss the collection of our NER dataset, including labels used and effort required. Then in Section 4.3 we present our full neural model for poetry-NER, and in Section 4.5 we present the performance of this model on our dataset. We conclude this chapter with discussion of future work and implications for traditional NER systems in Section 4.6.

## 4.1 Background and Challenges

Traditional named entity recognition datasets are extremely clean: they are pre-tokenized into documents and sentences. The CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003) is a good example of such a task: the goal is to compare learning algorithms rather than pre-processing steps, although feature engineering often includes pre-processing for NER.

We have the opposite data available: we know that a page contains poetry (given our identification model from the previous chapter) but we do not know which exact tokens on the page are poetry. Our scanned-books domain also adds additional constraints on our learning and strategies.

### 4.1.1 Background

Traditional entity recognition on news datasets uses a 4-class system; the fifth class is considered non-entity. This system was designed to be of use for news data, identifying the people, places, organizations and other entities critical to a news story.

**PER** refers to a person.

**LOC** refers to a location at any granularity: a building or a country.

**ORG** refers to an organization, like an assembly, a congress, a business or a religious organization.



**MISC** is a catch-all *other*, but in existing NER datasets it is mostly used for religious texts, e.g., “The Bible” or nationalities “English”, “German”, etc.

**O** is typically the negative class and the most prevalent class. NER is a classic example of an unbalanced learning problem, since most words and spans deserve this label.

Named entity recognition is not a novel problem, and there are a wide variety of pre-built solutions available. We initially explored using a variety of NER systems on poetry data, and found lackluster results. Since we initially believed the lack of formal news format was the cause of our issues, we explored using taggers trained to be effective on noisy, social media data.

#### 4.1.1.1 News-Based NER on Poetry Data

We present an example of a poem for which we have confirmed that traditional NER systems result in poor recognition in Figure 4.1.

Not a sup?–not a bite! Oh, why will temptation  
**Keep** trailin’ me up! Get out of my sight!  
**Or I** swar by my soul there will be a sensation,  
**And I** will get grub in the cooler to-night!  
**What’s** that? You know me of old? You’re another!  
And, hang you! if I wasn’t weaker’n water,  
**I’d**– What!– Git out!– You!– You, Ned!– My brother!  
**I** reckon I’m crazy, and that’s what’s the matter!

**Figure 4.1:** A stanza of a poem printed on page 48 of “The Poet Scout” (Crawford, 1886). The **bolded spans** were identified as person entities, and the *italicized span*. “**Keep**” was identified as a miscellaneous entity.

We note the lack of traditional capitalization even though punctuation is fairly regular in this stanza of the poem. The Spacy NER system is unable to identify instances of the person “Ned” on this page, and labels many capitalized words as “MISC” even when they are not entities. This shows that the domain of poetry is much different than that of news - where NER taggers perform quite well. Here, we were able to feed in the stanza itself to the tagger,

manually, to determine if any additional text on the page (page numbers, headers, etc.) was causing negative results, but we did not observe an improvement in tagging. Results here use the manually cleaned data.

Ultimately, existing tools were not helpful to us or our annotators (viewing such noisy tags led to more confusion than benefit). However, lessons learned from this research did inform the development of our model.

#### 4.1.1.2 Twitter-Based NER on Poetry Data

Given the failure of off-the-shelf NER systems for poetry data, we initially hoped to find an off-the-shelf system targeted at Twitter or other social media sources that might perform better on our noisy domain. However, as we will discuss in future sections, the challenge is not necessarily just noise from the OCR systems.

Additionally, Twitter models are trained for a variety of different classes and use extremely twitter-specific tokenization tools<sup>1</sup> (Ritter et al., 2011; Gimpel et al., 2011; Owoputi et al., 2012).

Instead of using these systems directly, we decided to explore using their *data* and their *labels* in a multi-task learning setup where our data and labels could stay the same (i.e., we would not need a COMPANY class or hashtag/URL parsing). We integrated using NER challenge data developed from twitter content, the W-NUT16 twitter NER dataset, provided by (Ritter et al., 2011)<sup>2</sup>. The W-NUT16 challenge provides data for more noisy named-entity recognition and tagging on social media (Strauss et al., 2016).

As an example of why this data was not directly useful for our purposes, the W-NUT16 data contains labels for “O”, “person”, “other”, “geo-loc”, “company”, “facility”, “product”, “musicartist”, “sportsteam”, “movie”, and “tvshow”. Due to the much older nature of our

---

<sup>1</sup><http://www.cs.cmu.edu/~ark/TweetNLP/>

<sup>2</sup>[https://github.com/aritter/twitter\\_nlp/tree/master/data/annotated/wnut16](https://github.com/aritter/twitter_nlp/tree/master/data/annotated/wnut16)

data, the modern “movie”, “tvshow” and even “sportsteam” are less likely to be relevant to our dataset.

### **4.1.2 Challenges**

In this section, we discuss the challenges that explain why traditional NER systems are presenting poor results on our data. These challenges inform our focus in developing our own NER models and how we collected our dataset of labels.

#### **4.1.2.1 No Sentences**

Poetry is built from a variety of structure, such as couplets, stanzas or free-form. As a result, there are typically no periods or obvious sentence boundaries. However, almost all off-the-shelf NLP/NER systems begin processing by first tokenizing and splitting into sentences. This is a step that makes little sense on poetry data and can result in very long “sentences” that lead to inefficient processing.

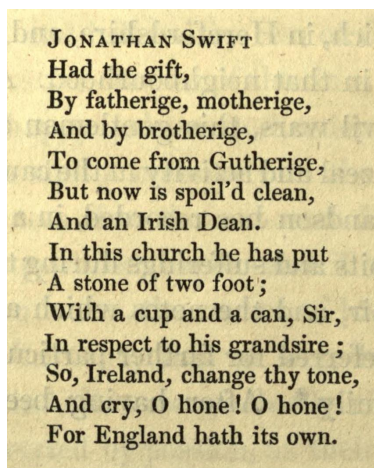
We originally designed a model based on sliding windows of a fixed size but this design made evaluation tricky (multiple predictions for the same positive or negative span needed to be combined). We later switched to feeding entire pages to our model (with line breaks as an explicit token), which resulted in more efficient and simple training.

Therefore, while collecting labels, we took care to label every token on a page that contained poetry; half-labeled examples could be confusing to any machine learning model, and so we labeled prose when it was co-located with poetry.

#### **4.1.2.2 Alternate Capitalization Patterns**

We know from our work on poetry identification that capitalization is useful for our formatting model (§3.2.1) but this is because the capitalization used in poetry (or lack of it) varies greatly from what we expect in prose - the traditional domain for named entity recognition.

While capitalization may still be a useful feature, it will be used in a different manner than in traditional text, where a word being capitalized is often a very strong baseline for entity recognition. This challenge is probably the reason that traditional NER taggers tried to label the first word on every line of many poems.



**Figure 4.2:** A poem about Jonathan Swift in a book composing some of his memoirs and notes (Swift and Scott, 1824).

Figure 4.2 contains a poem about Jonathan Swift. Although this poem contains periods as punctuation, it capitalizes each line, and these sentences are not fully grammatical, another challenge for traditional statistical and rule-based NER systems. Still, to a human, it is obvious that “England” and “Ireland” are locations, “Irish” may traditionally be marked as MISC, but “Irish Dean” is a reference to his profession and so might be modeled as a PER tag. “Gutherige” is most likely a reference to a school or another LOC, but is harder to tell from context (and without deeper research).

#### 4.1.2.3 Boilerplate text: HEADER

In digitally scanned books, we don’t know exactly where the boilerplate on a page is located. The first line of a page might be the title of the book, the title of the chapter, the name of the author, the page number or some mixture of the previous. It may also merely be the first line of content. We annotated non-content portions of the book with a

“HEADER” tag in order to see whether it would help our model learning in the way it helped our annotators focus on the important parts of the page.

Running headers interrupt sentences and poems as well as individual hyphenated words. Although we present the first work on poetry identification, we wanted to collect a dataset that would be aware of this challenge as results improve on the core task. A model that understands header separation from content would be useful for other tasks in this domain, such as page-number identification and static document enrichment for indexing.

#### 4.1.2.4 Non-traditional Entity Usage

Personification is a technique in poetry where concepts, animals or other non-human entities are given qualities of people. This is especially common with concepts like “life” and “death”, as shown in the excerpt below:

Death! death! death!  
Thou art both joy and ill;  
Death! death! death!  
Thou art both friend and foe!

These are the final four lines of a poem titled “Youth and Age” – two concepts that are also personified throughout, in addition to death (Donaldson Jr., 1860). While “death” is discussed as a joy and ill, it is also called a friend and foe – relationships that are typically reserved for people. Does this mean that the repeated “death” tokens should be labeled PER by a successful system?

We took a liberal stance on personification – if it was clearly personified, we labeled tokens as PER tags. In future labeling tasks, we might want to assign a specific label to personified objects that are distinctive from humans referenced as persons in the text which may improve accuracy or provide additional challenge.

Religious texts contain a fair number of more tricky classification questions. Is “God” an entity? If so, is “God” a person or a MISC? Is “Kingdom of God” a LOC that contains a PER? Is the “Archdiocese” a LOC or an ORG? In cases like this, we deferred to labeling

such spans with both appropriate labels, being as detailed as possible and with the idea of giving our models *partial credit* if they could identify either aspect of an entity.

## 4.2 A Dataset for Named Entity Recognition in Poetry

### 4.2.1 Page Candidate Selection

In order to evaluate named entity recognition on a dataset of poetry, we had to collect labels on a dataset of poetry. We chose to use our positive labels from our identification effort as seeds (§3.3.2) in order to minimize wasted effort, although we would occasionally label pages that were adjacent to those positively labeled if it were clear that a poem was broken across multiple pages.

### 4.2.2 Poetry-NER Token Classes

We labeled traditional NER classes with the goal of being comparable to existing systems, and added two of our own. While we prioritized labeling pages that our annotators had identified as poetry (Chapter 3) we also labeled prose and other boilerplate fully. That is people mentioned in text surrounding poems were also labeled. Therefore, our algorithms could take in full pages as input and not worry about some of the page being unlabeled.

**PER** refers to a person or personified object in poetry that is treated like a person.

**LOC** refers to a location at any granularity: a building or a country.

**ORG** refers to an organization, like an assembly, a congress, a business or a religious organization.

**MISC** is a catch-all *other*, but in existing NER datasets it is mostly used for religious texts, e.g., “The Bible” or nationalities “English”, “German”, etc.

**O** or **PROSE** is typically the negative class “O” in NER annotations, but for us it meant prose text.

**POETRY** was created to give us the ability to collect precise poetry extraction data while annotating individual tokens.

**HEADER** was created to indicate boilerplate on the page, specifically headers and footers. (§4.1.2.3)

### 4.2.3 Dataset Overview & Baseline Performance

When we finished collecting our dataset, we had collected nearly 6,000 tags across 600 pages that had been labeled as poetry by our annotators in our earlier tasks (Chapter 3). This sounds small in comparison to the full dataset we had available for identification, but labeling for NER is much more expensive, as we will detail in the next section, and it turns out that our dataset sizes compare favorably to other NER datasets (Table 4.1).

Measure		Poetry	CoNLL 2003	W-NUT16
Overall	Source	Poetry Pages	News	Twitter
	Unique Terms	27758	23865	14870
Labels	Example	Page Text	Sentence	Tweet
	Count	6	4	10
	PER	Yes	Yes	Yes
	LOC	Yes	Yes	GEO-LOC
	ORG	Yes	Yes	COMPANY
	MISC	Yes	Yes	OTHER
Stats	Splits Used	Train / Test	Train / TestA	Train / Dev
	Unique Terms	23152 / 9590	21009 / 9002	10579 / 6255
	Examples	493 / 138	14041 / 3250	2394 / 1000
	Example Length	233 / 261	14.5 / 15.8	19.4 / 16.2
	Mean Tags per Example	9.4 / 8.7	2.4 / 2.6	1.0 / 1.1
	Total Tags	4613 / 1196	34043 / 8603	2462 / 1128

**Table 4.1:** NER Dataset Statistics. This table describes the parameters of our novel Poetry dataset in the context of a news corpus (CoNLL 2003) and a microblog corpus (W-NUT16). Our examples are at the page-level and are much longer than sentences or tweets. Our test corpus for CoNLL is “testa”. Although our dataset is small, it compares favorably with CoNLL in terms of unique terms, and W-NUT16 in terms of total tagged words.

Although it is rather unfair to compare to off-the-shelf tools as we discussed previously (Section 4.1), we run an existing system in order to demonstrate the novelty of this dataset.

Spacy (Explosion AI<sup>3</sup>) is an efficient, modern NLP toolkit based on word embeddings and available in many languages. We processed our poetry dataset with this toolkit and record the results in Table 4.2, which are little better than random.

	P	R	$F_1$	AUC	Total
O	0.963	0.887	0.923	0.528	145215
PER	0.033	0.038	0.035	0.509	2702
LOC	0.031	0.029	0.030	0.511	1326
ORG	0.004	0.018	0.006	0.504	338
MISC	0.016	0.134	0.028	0.529	1376

**Table 4.2:** Performance of the Spacy multilingual model on our Poetry-NER dataset at the token level. Existing taggers are not better than random (AUC=0.5) on poetry data. Taggers are typically expected to be re-trained for new domains.

Note that results are not directly comparable to our later results because this represents our full dataset, without train/test splits. Existing tools do not necessarily provide the ability for retraining, but also depend upon specific pre-processing steps, such as sentence segmentation, which are not possible in Poetry.

#### 4.2.4 Labeling Effort

In this section, we estimate the economic cost of constructing this dataset. Once again, we collected timestamps when labels were submitted to our labeling system, so we can estimate the time it took any annotator to submit a single label.

A single tag took a mean of 10.5s and a median of 5s to submit. There is a mean of 10.7 labels on each page and a median of 7 labels per page. This meant that a fast-average page might be (5s · 7 labels) = 45s and a slower-average page might be (10.5s · 10.7 labels) = 112s or nearly 2 minutes. Note the numbers here include the full set of tags, including MISTAKE tags (the UI solution to deleting tags) and other preliminary classes eventually excluded from our analysis.

---

<sup>3</sup><https://spacy.io/> 2018



Unfortunately, we did not collect user dwell time. We know annotators spent a lot of time reading before entering labels, but our interface was not configured to log and collect this delay, so our time calculation only includes time spent entering labels; we suspect that actual labeling time was much longer (2-3x) more than we present in this section.

Our dataset of around 6,000 tags was collected in an estimated 17 hours of expert annotation. Collecting overlapping labels in a sophisticated user interface required much more work and deep inspection of the original text in order to make sense of the content than our more-simple poetry/not-poetry labeling from the prior chapter (§3.3.2.6). We had annotators completely label entire pages, even if poetry was only available on a small section of the page, and to mark the actual token-boundaries of the poetry itself.

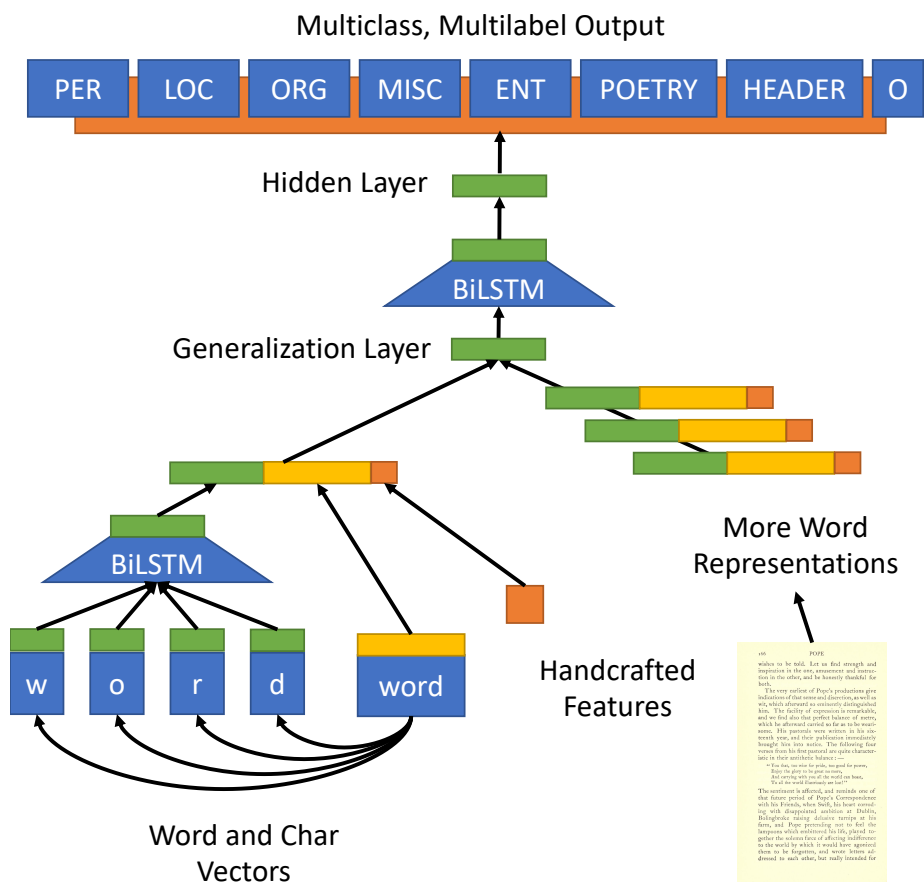
Here, our labeling effort estimates are clearly underestimates. It was typical during annotation to pull the actual pages from the book, to read the poem on the left and right of the random page surfaced and then to begin annotating once we were pretty sure of the results. Due to alignment issues, an early set of labeled documents had to be removed. Annotators also occasionally made mistakes and those labels had to be deleted, so the timing from that work is lost.

### **4.3 Poetry NER Model**

In this section, we give an overview of the features and components that we used to build our Poetry NER model. A graphical representation of our model is available in Figure 4.3, and we study the relative benefit of these components in Section 4.5.4.

#### **4.3.1 Sequence Prediction Model**

Named entity recognition is a classical sequence prediction task. Although we do not segment our document into sentences, our task is still a sequence prediction model. We feed whole pages into our models, including tokens that represent punctuation and line breaks. In a way, we are moving traditional pre-processing into the learning algorithm and letting the



**Figure 4.3:** Graphical representation of our Poetry-NER model. Word representations are built from handcrafted features, an LSTM of character embeddings and word embeddings. This is generalized and sent through another LSTM before going through hidden layers to our multiclass and multilabel predictions.

algorithm decide when it is appropriate for poetry, given that poetry does not necessarily have semantic sentence boundaries or line breaks.

The core of our model is a bidirectional long short-term memory or BiLSTM layer. These layers are now standard in NLP and IR tasks, and performed best among sequence adapters we explored in Chapter 3 for content-based poetry identification. These layers walk over the tokens in the input page in a left-to-right and right-to-left manner, accumulating a hidden state which is then used as a per-token output. We add another dense layer atop this output, and then use a multi-class logistic regression layer to predict the token label for each term.

In future work, we hope to explore conditional-random-field models as an alternative output layer, adding additional dependencies between tokens for more accurate prediction. We focus on the feasibility of this task and quantifying the importance of the other model structures that are common in modern NER models.

### 4.3.2 Word Representations

Named entity recognition systems often include several sources of representations for words. We use both character-LSTM vectors and handcrafted features to represent “sub-word” information and to aid with out-of-vocabulary errors, we include externally trained word embeddings<sup>4</sup>, and feed our output into a generalization layer before feeding those representations into a sequence prediction model.

Because of the high cost of acquiring NER data for poetry (§4.1.2.4 and §4.2.4), we still have a relatively small dataset with which to train an effective tagger. Since we desire to generalize outside of the 600 pages we labeled, we need to be careful not to derive our own vocabulary set: the vocabulary in our training data is necessarily a small subset of the vocabulary in the interesting set of poems we are able to collect.

#### 4.3.2.1 Character-LSTM Representations

Constructing a comprehensive set of every character in English is straightforward: ignoring punctuation, there are only 26 such characters, or 52 with capitalization. Adding in Unicode characters makes things a little more tricky, with  $2^{20}$  available code-points across all encodable printed languages, but in practice books are not available in every Unicode language due to a lack of optical-character-recognition software.

Despite the domain of characters being a rather small space, character embeddings trained alongside word embeddings for NER have been shown to achieve state-of-the-art results

---

<sup>4</sup>300 dimension GloVe vectors trained on 42B tokens of the Common Crawl (Pennington et al., 2014).

without handcrafted features (Lample et al., 2016). We explore a character-LSTM as input to our NER system for this reason.

### 4.3.2.2 Handcrafted Word Features

We include a small set of designed word features. Although one of the great advantages of neural architectures is their ability to learn what features are important, this typically requires a large amount of training data and we have only a modest quantity of training data (Table 4.1).

Name	Type
Empty	Boolean
First-Capitalized	Boolean
Punctuation	Boolean
Newline	Boolean
Hyphenated	Boolean
Digits-Fraction	Double
Capitalized-Fraction	Double
$\text{sigmoid}(\text{len}(\text{Token}))$	Double

**Table 4.3:** List of Handcrafted NER Word Features

We created eight simple handcrafted features to deal with limitations of our pre-trained word embeddings (our GloVe vectors did not have uppercase words, punctuation, or newlines) and to expand our representation. These features are presented in Table 4.3. In theory most of their information could all be inferred by the Character-LSTM representation we designed in the prior section, but designing these features explicitly help increase the power of our model with our limited dataset.

Some of these features are explicitly designed to target header sections (which are sometimes completely capitalized) and others were inspired by our poetry-identification model.

### 4.3.2.3 Simple Attention

We additionally experimented with an additional feature, Attention, which was designed to encode the position on the page in a relative manner, inspired by the positional encoding used in transformer works (Vaswani et al., 2017).

These networks typically include sinusoidal features with very large periods so their recurrent units (our LSTM) may learn relative time differences between words in longer text.

$$\text{SimpleAttention}(t) = \sin\left(\frac{t}{10000}\right)$$

This feature was simple to calculate, where  $t$  is the integer index position on a page, and provides some small portion of modern attention models.

### 4.3.2.4 Word Generalization Layer

Given our set of word representations, we construct a unified representation by concatenating the vectors for each word: the character-LSTM output, the handcrafted word features, and the pre-trained GloVe embedding. We then add in a *generalization* layer which maps from this large space ( $300 + d_g + 9$ ) to a much smaller space ( $d_i$ ) for input into the word-at-a-time LSTM (where, e.g.,  $d_g = d_i = 32$ ). Actual layer sizes were selected as hyper-parameters.

This layer allows for the network to learn which interactions in all the word features are important before combining with other words in the local context. It also allows for making the word embeddings more task-specific while preserving the ability of them to generalize to the larger, unlabeled datasets from which word embeddings can be trained.

## 4.4 Experimental Setup

In this section, we discuss details of our experimental setup not previously mentioned. We designed our NER models using the PyTorch library<sup>5</sup>. In order to have reproducible

---

<sup>5</sup><https://pytorch.org/>

results, we ran many iterations of each model (Section 4.4.1), we leverage AUC to avoid balancing issues in the dataset (Section 4.4.2) and we frequently present performance on a unified, summary “ENT” class rather than fit four curves on every chart (Section 4.4.3).

#### 4.4.1 Stochasticity and Variance

We note that our studies are made more difficult by the stochastic nature of modern neural NER models. Since network weights are initialized from random distributions and one of the most effective models for regularization is randomized dropout (Srivastava et al., 2014), training the exact same model over the same data in the same order will lead to slightly different predictions over time (not to mention that randomly shuffling data per-epoch is critical to effective learning).

Another issue encountered was one of training time. Early-stopping, or only training for a fixed number of epochs is a key technique for regularization in neural networks, but fixing a number of epochs to train meant that when changes made it more difficult to learn with the same amount of training, the final models appeared worse. We considered this a reasonable trade-off for most of our experiments and trained for a fixed 60 epochs for most experiments. We inspected training logs to ensure that our models appeared to be converging in the last few epochs but know of no methods to ensure this behavior.

In order to limit these problems, we used consistent random seeding, we froze a testing dataset and and chose measures that were most stable to evaluate.

Recently, more focus has been put on the importance of evaluating neural models correctly, and the high variability of model performances has been exposed in multiple domains (Cohen et al., 2018; Clary et al., 2018). Our study of features (Section 4.5.4) will look therefore look at the ranking over all seeds rather than just the means of models merged together to fully present the variance in a non-parameteric manner.

#### 4.4.2 Measure Selection

Traditional works on named entity retrieval use instance-level precision, recall, and  $F_1$  - a harsh measure where tags are only considered accurate if they are precisely located - overlap is not enough. Since we are targeting a challenging domain, we focus on token-level measures, a little more forgiving, as labeling part of someone’s name as an entity receives some credit. A token-level loss is important for learning - we want models that are closer to success to appear that way.

We initially used set-based precision, recall, and  $F_1$  as our measures for this task at the token level, but found that our models were not particularly good at selecting cutoff values that generalized well. This is a challenge of having a small dataset, and it made comparisons difficult – sometimes models would learn great cutoffs and their  $F_1$  would appear to be much higher than another model, but we could not tell if that was because of the item under test or noise in the cutoff selection. Also, since our classes were of different sizes and prior probabilities, it was hard to tell if the model was performing well, or learning “YES” or “NO” classifiers. Although this problem became less frequent as our dataset grew, it makes analysis of how many labels were required difficult.

For this reason, we focus our analysis on area under the ROC curve (AUC) - a measure that compares true positives and false positives under the ordering provided by the classifier, and has a natural comparison to random noise: an AUC of 0.5 means the tagger is providing less information than random, regardless of test set size, prior probability of the class being analyzed, or cutoff learned. AUC is effectively a ranking measure.

In all analysis, we ignore the positive benefits of averaging in the prediction of the negative “O” class, which was for prose in our dataset, since this class dramatically outnumbers all other classes in averages and inflates performance. AUC is not a traditional measure choice for NER, but makes us more confident in our analysis of the power of features with a more robust measure.

### 4.4.3 Multiclass Weighting and the ENT class

Since we treated NER as a multi-class labeling problem, we needed to weight tokens as instances in a loss function. We chose to assign weights to classes based on their prevalence in the dataset. We leveraged the “balanced” heuristic present in `sklearn`<sup>6</sup> which is attributed in documentation to a work on logistic regression over rare events by King and Zeng (2001). The weight  $w_k$  of a class  $k$  is based on its frequency in the training set labels  $y$ .

$$w_k = 1.0 - \frac{|y_k|}{|y|}$$

Because our annotators found labeling poetry entities to a single class difficult, we are actually evaluating a multi-class multi-label problem. We therefore introduced a meta-class of “ENT” in our training data, to explore whether individual classes (PER, LOC, ORG, MISC) were meaningful.

Because we formulated our tagging problem as multi-label and multi-class, including an ENT label that is algorithmically assigned to any PER, LOC, ORG, or MISC tags creates a sort of hierarchical loss. While the classifier cannot pinpoint which specific class a span of tokens should be, it can still earn performance points by noticing that those tokens are some kind of entity.

## 4.5 Results

In this section, we present results from our study of named entity recognition models on poetry.

We start by evaluating our experimental assumptions. Section 4.5.1 shows results broken down by NER label and how they correlate with our summary class. In Section 4.5.2 we validate that returns from collecting more labels have begun to diminish, and our dataset is

---

<sup>6</sup><https://scikit-learn.org/>

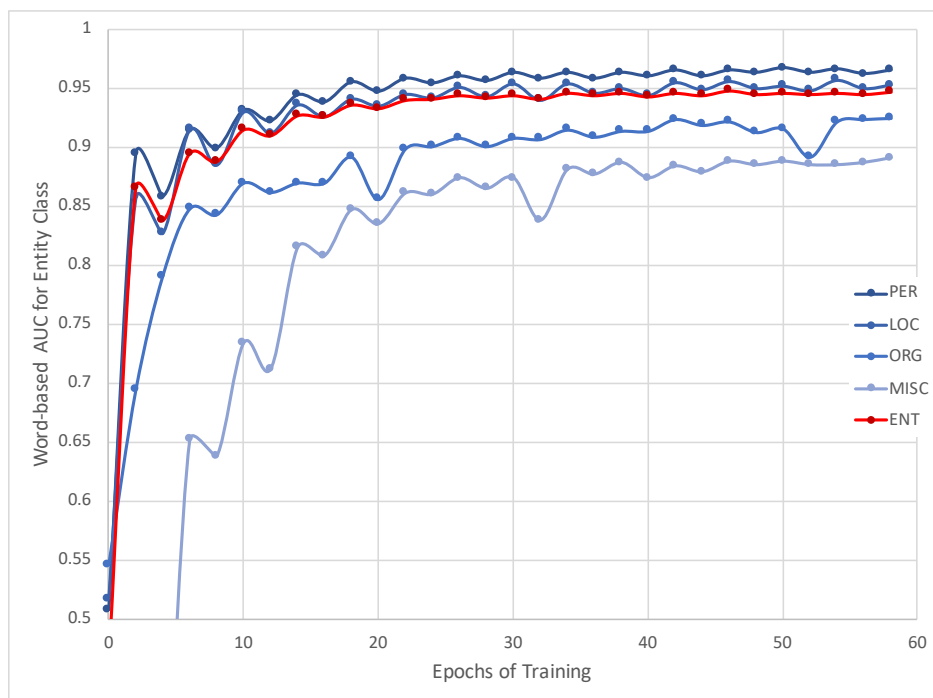


sufficient for reasonable evaluation of models. Then we look at performance of non-entity labels collected: the most promising being POETRY and HEADER (Section 4.5.3).

We close by performing a deeper feature-ablation study, looking at pieces of our model as well as cross-training on news and social media datasets in Section 4.5.4.

### 4.5.1 Validity of ENT Summary Class

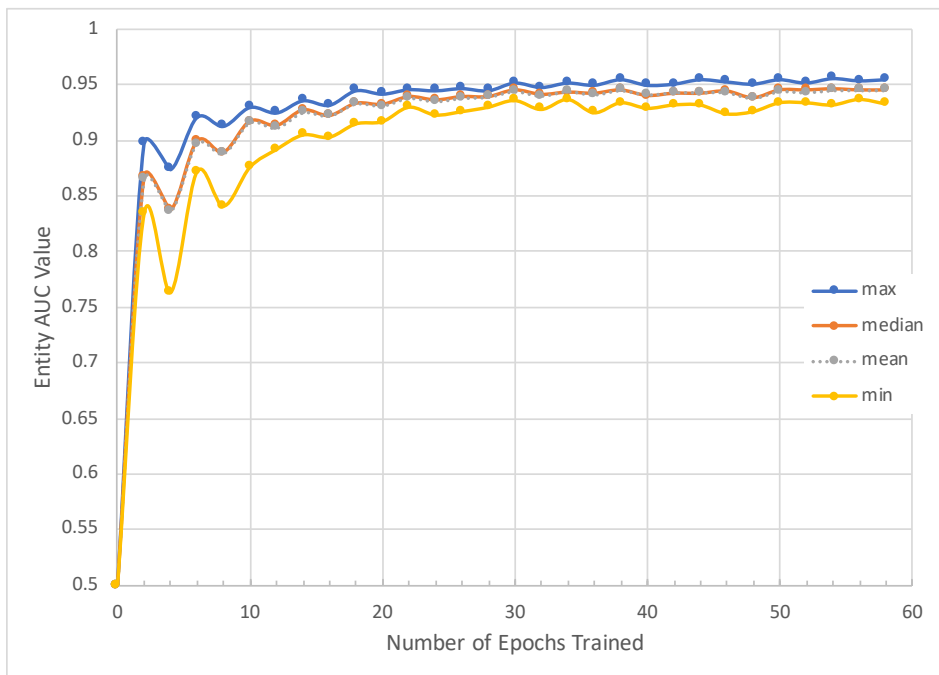
In Figure 4.4, we present the learning rates of the individual entity classes. This figure clearly shows that MISC is the most difficult entity label, and it takes approximately five epochs of training before our average-seed NER taggers become capable of recognizing it with better than random AUC.



**Figure 4.4:** Mean Entity, PER, LOC, ORG and MISC performance across 30 trials and trained for 60 epochs.

While our the presence of our ENT class (and its effect on multi-class loss) does not provide a significant improvement: noted as “Separate Classes” in (Figures 4.8 and 4.9) it provides a useful way to discuss whole models together, provided that one trains long enough to ensure that rarer classes, such as MISC, are performing well.

Figure 4.5 presents the distribution of the ENT performance across 30 trials, we find that mean and median are very well centered, and apparently little variation, given the tightness of Maximum and Minimum plots. We believe performance appears to oscillate every four epochs, due to the internals of the Adam optimizer manipulating the learning rate (Kingma and Ba, 2014), since it is consistent across shuffles and randomly-init neural networks and then diminishes with time.



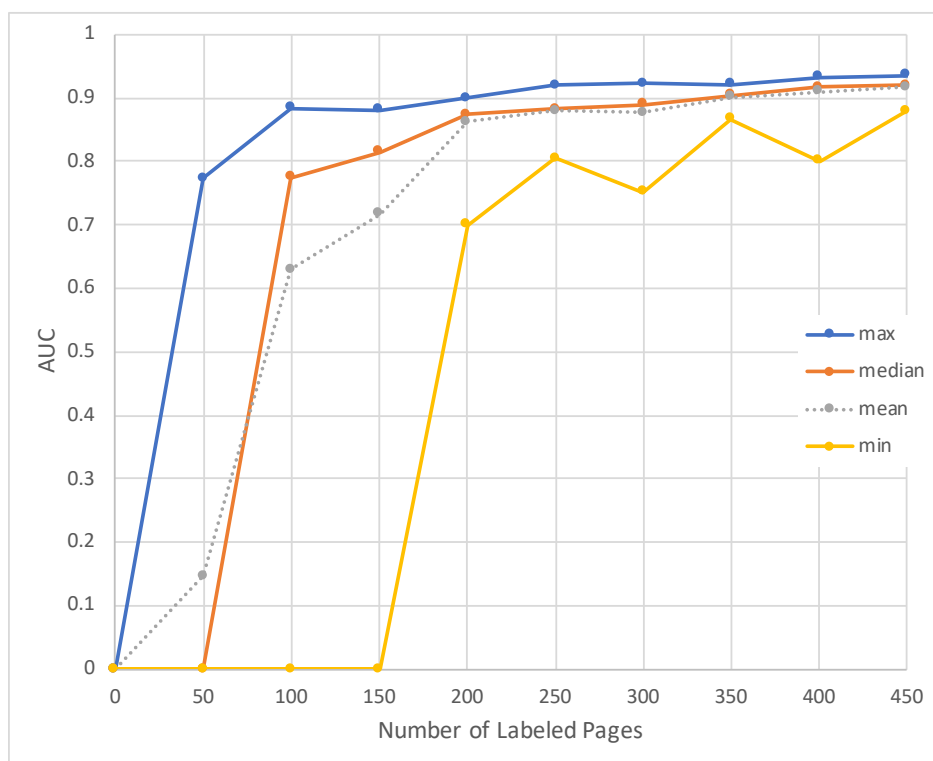
**Figure 4.5:** Mean, Median, Maximum and Minimum Entity performance across 30 trials and trained for 60 epochs.

#### 4.5.2 Label Quantity Study

We also study whether the quantity of labels obtained for our NER task is sufficient for observations about our task to be meaningful. It is expected that over time there will be diminishing returns on performance, where results will plateau due to the quality of features, the inherent difficulty of a task, or the power of the model used and adding more labels to a classifier of any kind will cause minimal improvement.

Figure 4.6 presents our results at increments of 50 labeled pages up to a maximum of 450 in the training set. It takes until about 100 pages labeled before our average model begins to perform better than random, and 200 pages labeled before our worst model will perform better than random.

Since labeling 50 pages takes a few hours of annotation time, we decided that this was an acceptable stopping point. We expect that doubling the number of labels available will be required to see dramatic improvements.

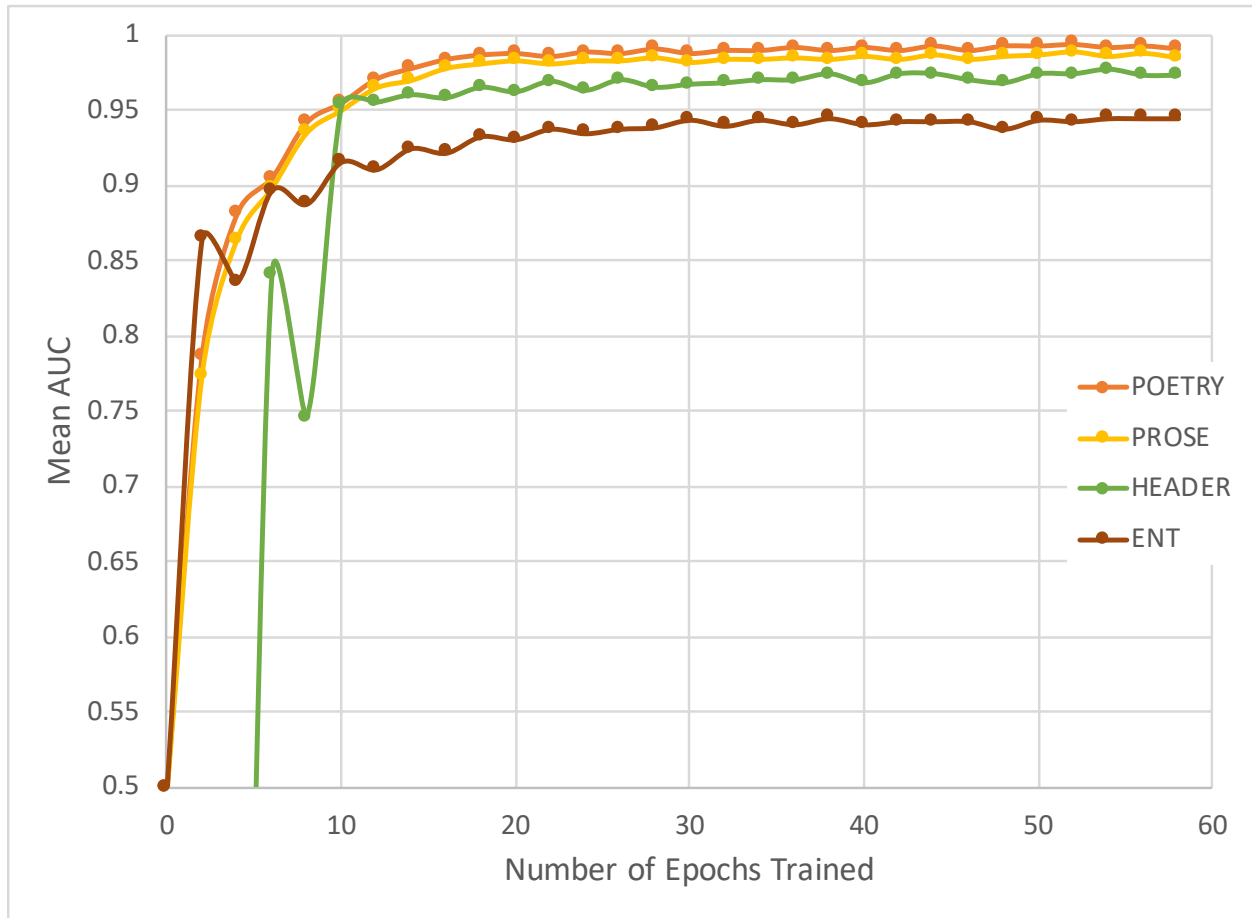


**Figure 4.6:** Mean, Median, Maximum and Minimum Entity AUC over 30 trials trained for 60 epochs with varying sizes of training data.

### 4.5.3 POETRY, PROSE, & HEADER Detection

Although our primary goal was to detect entities within identified poetry, we included a HEADER class in our labeling UI to identify boilerplate (e.g., titles, authors and running headers/footers in books). Figure 4.7 shows just how promising the token-level approach to header, poetry and prose detection is based on this architecture. In future work we plan

to investigate expanding our poetry identification (Chapter 3) to the token-level using this model.



**Figure 4.7:** Performance of HEADER, POETRY, PROSE and ENT classes.

#### 4.5.4 Feature Ablation Study

We briefly summarize the different parameters we explored as “features” in our NER model. For full descriptions, please refer back to Section 4.3. For their mean performance, please refer to Figure 4.8. For an impression of the variability of these options, please refer to Figure 4.9.

We review the features below (and their chart notation) in order of their mean performance.

**+Attention (0.948)** We added a simple sinusoidal positional feature to capture a small portion of modern attention models. Although there was slight benefit here, we are hesitant to include more sophisticated portions of attention models in such a small dataset.

**+Twitter (0.947)** The W-NUT16 dataset was available for cross-domain training at every training step, using a separate output layer, but sharing LSTM and word embeddings. A very slight benefit was observed from this feature as well, at the expense of using a lot more data to train.

**Separate Classes (0.946)** Whether the full model was trained with PER, LOC, ORG and MISC classes, whose individual performance was generally lower due to the inherent ambiguity in NER classes. There might be some benefit to having the full information, but it is not clear.

**Full Model (0.945)** The full model contains all of our engineered features and layers (as well as training with CoNLL data).

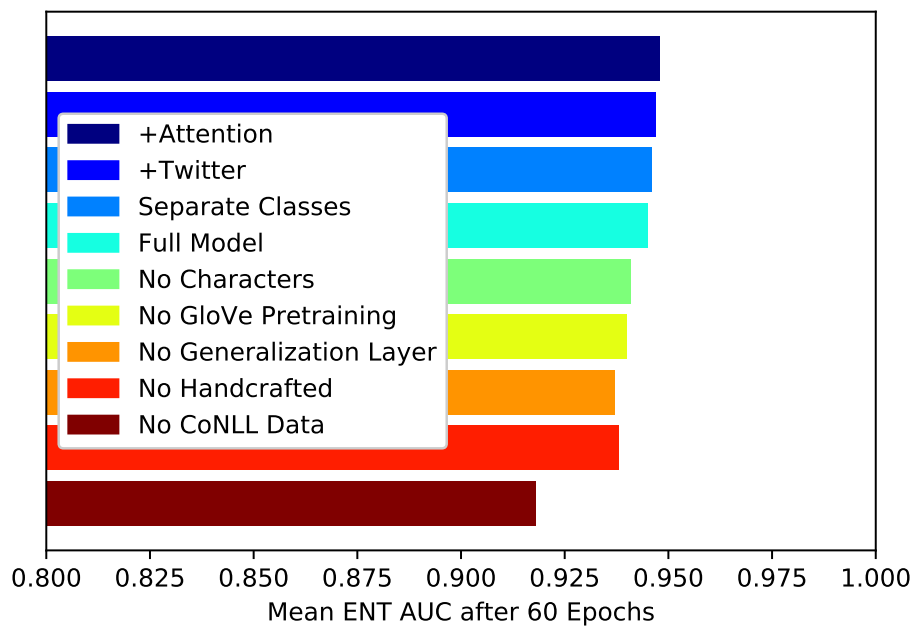
**No Characters (0.941)** This model is the same as our full model, but without character embeddings or a character LSTM. This model trains much faster, but at the expense of some performance on average.

**No GloVe Pretraining (0.940)** This model is the same as the Full Model does not use GloVe word embeddings, but only learns embeddings for words observed during training. It does have character embeddings and CoNLL data from which to generalize, but it is also weaker.

**No Generalization Layer (0.937)** Removing the layer between our character features and the LSTM makes it harder for the network to learn which features or combinations of features are important.

**No Handcrafted (0.938)** Without our handcrafted features, it falls on the character-LSTM model to learn specific, useful features, such as capitalization.

**No CoNLL Data (0.918)** By far, the most impactful loss on our dataset was removing CoNLL data from training. Although existing news taggers do not perform well on our data, there is definitely similarities in the task and having more, clearer examples of our NER classes definitely benefitted our models.



**Figure 4.8:** Means of each feature setting over 30 trials trained for 60 epochs.

Due to the high variance in trained models, in addition to presenting results on the mean models trained under each setting, we also present a graphical representation of the ranking of our models. This ranking plot is attempting to visualize a form of significance testing (Figure 4.9).

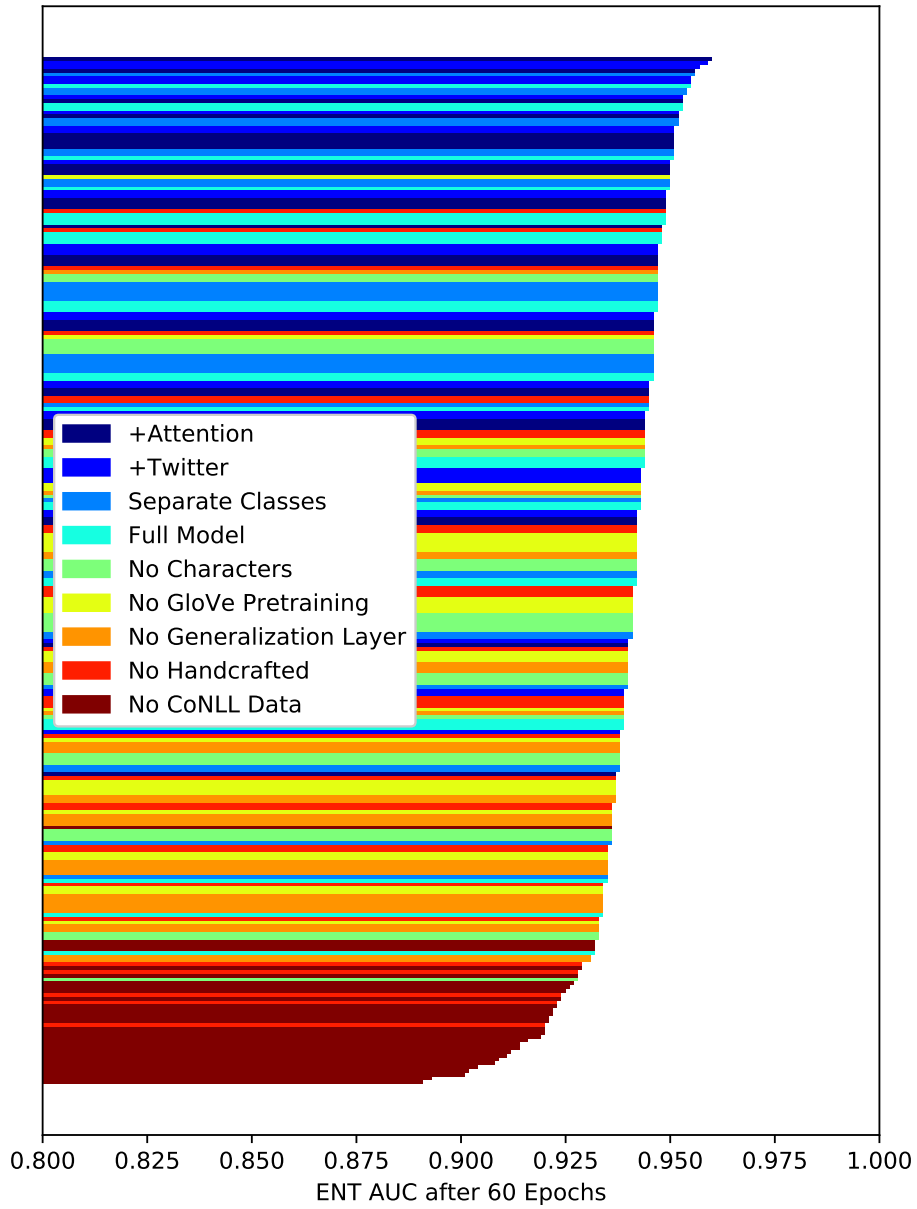
Seeing that so many methods are interleaved, the conclusion we feel most confident about is the importance of CoNLL data.

#### 4.5.4.1 The Importance of CoNLL Data

The CoNLL data was critical for learning entity representations in our poetry dataset. As a larger, cleaner dataset, it allowed our model to quickly learn what kinds of words are entities and what kinds of words are not. Because we provided few features beyond the word vector itself, this intuition carried well across domains.

In fact, without mixing the CoNLL data into the poetry dataset, our architecture spends many epochs of training struggling with entity annotations and minimizing loss solely based on the easier “O” (prose) token classes.

In addition to the CoNLL03 data providing a strong “jump-start” to learning and achieved strong performance after just a few minutes, training our models for much longer still resulted in a moderate benefit (Figure 4.9). We therefore found CoNLL03 useful for both pre-training but not actually for capturing some additional diversity.



**Figure 4.9:** Ranking-based bar plot of all 30 trials of neural training for our feature explorations.



## 4.6 Discussion

We close this chapter with a discussion of remaining challenges for future work and implications of our findings for traditional tasks.

### 4.6.1 Remaining Challenges for Poetry NER

Although we have some confidence we have labeled enough data to properly evaluate our task, one of our key findings in this thesis is that poetry is an extremely broad domain. In order to have confidence in poetry entity-recognition we will need to collect a broader set of labels from a larger quantity of books.

Because of the challenge of placing traditional entity labels on subjects in poetry (which are often personified) we believe that it would be interesting to explore knowledge-base driven approaches to NER as done in some prior work (Foley et al., 2016) and to also perform some kind of entity linking, incorporating a knowledge base as a kind of background knowledge to help contextualize entities like *Death* – poetry offers the dual task of identifying the entity type (perhaps MISC) and its role in the current line of poetry (perhaps PER).

We hope to leverage our per-token poetry labels (Figure 4.7) collected to explore integration with our poetry identification models in the future, so that more exact boundaries can be detected in our poetry collections.

In the future, we hope to expand our poetry datasets to better study the differences in time. Focusing on historical eras and changes in language will help us separate the effect of poetry on our NLP tasks from the historical variants of language. To improve from there, we expect to need word-generalization approaches such as the VARD word-variant replacement tool (Rayson et al., 2007; Baron and Rayson, 2008).

### 4.6.2 Implications for traditional NER

Although we did discover a modest boost while using traditional NER datasets in a multitask learning setup, this was only a small increase in AUC. This result suggests that the diversity available in poetry could be included in more traditional NER datasets in order

to provide a novel challenge to systems and researchers interested in this task without being counter-productive to any particular task performance.

The success of our multi-task learning setup is particularly promising in light of results presented by Bamman (2017), where part-of-speech tagging has struggled on historical and literary domains, and named-entity-recognition is known to generalize poorly from news to social media. Knowing that poetry is a broad domain from our results in chapter 3, some obvious future work here would be to perform other NLP tasks (such as part-of-speech labeling) and to collect data explicitly randomized by publication date in order to ensure generality.

Our analysis of some features added to our NER network suggest that a lot of useful handcrafted features such as the generalization layer (or actually handcrafted features) may aid in training speed performance more than they provide an overall gain (Figures 4.9,4.8). Giving neural networks sufficient time to make progress is important (Figures 4.54.4) as is training many seeds per model (Figure 4.9). More work would do well to discuss the practicalities of training these state-of-the-art models, and present more robust evaluations across more trials and random seeds for newly presented models.

We hope that interest in our poetry-NER dataset will lead to development of models and systems that more gracefully handle genre changes and understand both poetry and prose in appropriate context.

## CHAPTER 5

### POETRY RETRIEVAL

In this chapter, we define and explore the first poetry retrieval corpus and dataset. In information retrieval (IR), we refer to collections of documents as *corpora*, and typically reserve *dataset* for collections of queries labeled against those corpora.

Using our corpus of pages identified as containing poetry, we have the ability to retrieve only poetry in response to queries. Unfortunately, because we do not have a live system with users we cannot directly analyze our performance based on actual queries. Instead, we construct a dataset based on our insights from two sources: web query logs mentioning words related to poetry, and the categories that were manually labeled in the `poetryfoundation.org` dataset by contributing users. First, we analyze these two sources and present mostly qualitative results about how users have searched for (§5.1.1) or organized poetry (§5.1.2) the past.

In Section 5.2 we present a set of models for poetry retrieval. We have a selection of traditional retrieval models as well as two novel models based on categorical and emotional vectors. In Section 5.3 we present our new, open dataset for poetry retrieval, which is the first of its kind. In Section 5.4, we discuss the results of our model comparison.

In general, we find that query-expansion features represent promising directions for future research. Finding appropriate external resources will help us to improve poetry ranking systems in the future. Additionally, experiments with vectors derived from explicitly emotional data as well as poetry categories suggest that better understanding of emotions and categories in poetry definitely make a difference in ranking.

## 5.1 Analysis of Poetry Information Needs

In order to create a realistic test collection for poetry retrieval, we analyze some real world sources in order to generate meaningful and representative test queries.

### 5.1.1 Query Logs Analysis

The AOL query log contains 21 million unique queries (Pass et al., 2006). The MSN query log contains 15 million queries sampled over 1 month (Craswell, 2009). While these query logs contain millions of queries, not all of them are relevant to poetry. To focus our analysis, we first restrict ourselves to sampling queries that contain the terms “poetry”, “poem” or “poet”.

While this is not an exhaustive analysis, (e.g., we could compose a list of terms describing alternate forms of poetry, e.g., “haiku”, “sonnet” or even lists of poets and musicians) this allows us to quickly identify a large set of queries that are meaningful and relevant while being simple and consistent.

We combine the queries observed from these two logs containing our query terms in order to get a broader view of poetry search.

Although analyzing user sessions that mention poetry might be interesting future work, looking at user-specific data in these logs is ethically questionable, as it has been shown possible to de-anonymize a large number of these users (Amitay and Broder, 2008).

#### 5.1.1.1 Qualitative Results

We labeled the 200 most frequent queries in the concatenation of our two query logs that matched our poetry filter. All of these query strings occurred more than 27 times across the two logs. For comparison, the most frequent query was the unspecified “poems”, at 4,128 occurrences, which perhaps indicates an intent to browse popular poetry. By looking at common, real-world queries, we were able to draw some general observations about users’ poetry-seeking behavior.

As we labeled the most frequent queries in our dataset, we discovered the following fine-grained categories:

**Vague** queries were those searching just for poem or poems; on the web it is important to select the genre the user desires, but in a poetry dataset this query is uninteresting.

**Topic** queries tended to filter for queries on a particular subject, be that emotional “love”, or less so: “teacher”.

**Mood** queries seem different than topical in that the user has not specified the queries be about any particular topic, but have a certain tone: “inspirational”, “sad”, or “funny” fell into this category.

**Holiday** queries tended to explicitly mention a holiday; we guess that the information need here is a desire to select a poem for use in a greeting card or for reading at a gathering.

**Life-Event** queries are similar to holidays but are not tied to a particular calendar date: “graduation”, “funeral”, “birthday” and “new child” are all examples of this kind of query.

We present a view of the most frequent results, along with counts and labels in Table 5.1. Many of the top results are users searching for poems for a particular holiday or life-event, although topic-oriented queries such as “love” or “friendship” are also common. These are subtly different than searches for “sad”, “inspirational” or “funny” poems, which are looking for more of an emotion or genre than a particular topic (e.g., love poems could be funny or sad.)

As we moved further down the ranking, unfortunately it became more and more difficult to assign our fine-grained categories. For example, are “cheating”, “break up” or “broken-heart” poems topical or mood oriented? Our initial category here is topical, i.e., that an emotion is a topic. We found some additional queries aimed at specific poets, e.g., “langston hughes”, and developed a more coarse-grained but broader set of categories:

Query	Count	Label
poems	4128	Vague
love poems	2872	Topic
mothers day poems	966	Holiday
mother’s day poems	713	Holiday
friendship poems	418	Topic
easter poems	319	Holiday
graduation poems	315	Life-Event
inspirational poems	279	Mood
memorial day poems	251	Holiday
birthday poems	239	Life-Event
sad poems	211	Mood
mothers day poem	211	Holiday
poem	193	Vague
teacher poems	190	Topic
funny poems	182	Mood

**Table 5.1:** Most frequent queries including stemmed forms of {poetry,poem,poet} in raw, unstemmed format.

**Topic** became the preferred category of ambiguous Mood and Topic queries, as it was often difficult to distinguish intent; e.g., is “depression” a mood or a topic?

**Mood** continued as a label for queries that were purely emotional or aimed for a certain tone or mood.

**Metadata** became our catch-all for poetry searched by title, author, license, or other meta-category.

**Event** became a combination of our life-event and holiday categories, since the information need behind them were roughly the same.

**Other** queries included queries that appeared to directly quote famous poetry (e.g., “roses are red poems”), things that were not actually searching for poems (e.g., “academy of american poets”), like searching for analysis of famous poetry – perhaps to complete homework.

We present a distribution of these labels in Table 5.2. Note that the sum of unique queries is higher than 200 because we often gave multiple labels to a single query. Other is large because it includes the vague queries, e.g., “poems”.

Label	Unique Queries	Total Queries
Topic	149	10022
Metadata	78	3062
Event	59	4737
Mood	24	916
Other	17	4891

**Table 5.2:** Distribution of labels across the most frequent 200 queries.

Moving further down the list, as queries become more rare, we found more combinations of user intents. Some queries are both topical and emotion or mood-queries, like “silly poems about seasons”. Or topical and structural: “love poems in spanish”, or for an event and a mood: “inspirational mothers day poems”. Many queries also specified a target age demographic: “funny poems for kids”. All of these examples highlight a desire to refine, and filter results beyond a simple tag-based system to at least the inclusion of multiple tags. Some poems called out a particular religious denomination, e.g., “christian easter poems” while others simply requested any religion, “religious love poems”.

A large sample of queries is available, organized by category in Table 5.3. We can see that Mother’s Day and Father’s Day are big events in the dataset, and that queries for these events also get decorated with topical and event needs.

Many queries asked for poems by a specific author, poems of a particular style (e.g., acrostic), poems that were good for children, or poems that were license-free or of a particular length. These metadata challenges are potential avenues for future work, but require significant external data or inference about sources themselves that are out-of-scope for this work. Reasoning about the publication date and country of source documents would allow us to tag extracted poetry with licenses, but is not a core textual understanding task.

### 5.1.2 PoetryFoundation Categories

The `poetryfoundation.org` dataset has “keyword” category labels on some of the poems. Out of the 12,959 poems, 9,071 have at least one of 136 category-tags specified. These labels cover a wide variety of ideas and topics. We present the full alphabetic list in Figure 5.1. It may seem strange at first that “Poetry” is a category label, but not all poetry is about poetry. Upon seeing this category, we wondered if there were actually non-poetry within this dataset (Chapter 3), but the `poetryfoundation.org` dataset appears to only have documents that are poetry contributed by users, and this category therefore contains “meta-poetry.”

This set of categories is sparse, and some have few poems actually labeled underneath them, but they provide a more interesting and more topical set than the queries we found in our search logs. Each category has a mean of 461 poems and a median of 291 poems labeled for that category, out of the 12,959 poems. The most frequently and least frequently used categories are presented in Table 5.4

There are 25 categories with fewer than 100 poems labeled, and 12 categories with more than 1000 poems labeled. The other 99 categories are distributed in the middle. Each poem has a mean of 4.8 categories and a median of 4 categories assigned to it. The poem with the most categories assigned has 41 such category keywords. A full 3,888 poems (30.0%) do not have any categories assigned.

Since poetry as a medium is built to evoke emotion rather than provide knowledge, we suspect that categories will provide a useful method of providing serendipitous browsing and searching opportunities (André et al., 2009). Optimizing for serendipity has provided gains in user satisfaction in music recommendation (Zhang et al., 2012). Our desire to present these categories in some way motivates their choice as queries and as a retrieval model (§5.2.4).

The fact that both of these datasets are sampled from modernity may pose a problem. For instance, the concepts of “divorce”, “race”, “popular culture”, and “urban life” have changed drastically (as an understatement) since modern copyright laws went into effect in the 1920s and have been extended since. Certainly the poetry in the `poetryfoundation.org`



Activities, Ancestors, Animals, Architecture, Arts, Birth, Birthdays, Books, Break-ups, Brevity, Buddhism, Christianity, Cities, Class, Classic Love, Coming of Age, Companionship, Complicated, Conflict, Country Life, Crime, Crushes, Dance, Death, Design, Desire, Disappointment, Divorce, Doubt, Drinking, Eating, Economics, Enemies, Ethnicity, Failure, Fairy-tales, Faith, Fall, Family, Film, First Love, Flowers, Folklore, Friends, Gardening, Gender, Ghosts, God, Greek, Grieving, Growing Old, Health, Heartache, Heroes, History, Home Life, Horror, Humor, Illness, Indoor Activities, Infancy, Infatuation, Islam, Jobs, Journeys, Judaism, Landscapes, Language, Learning, Legends, Life Choices, Linguistics, Living, Loss, Love, Marriage, Men, Midlife, Money, Music, Mythology, Nature, Other Religions, Outdoor Activities, Painting, Parenthood, Pastorals, Patriotism, Pets, Philosophy, Photography, Poetry, Poets, Politics, Popular Culture, Punishment, Race, Reading, Realistic, Relationships, Religion, Roman Mythology, Romantic Love, Satire, School, Sciences, Sculpture, Seas, Rivers,, Separation, Sexuality, Social Commentaries, Sorrow, Sports, Spring, Stars, Planets, Heavens, Streams, Summer, The Body, The Mind, The Spiritual, Theater, Time, Town, Travels, Trees, Unrequited Love, Urban Life, Vexed Love, War, Weather, Winter, Women, Working, Youth, the Divine, the Supernatural

**Figure 5.1:** Alphabetized list of 136 `poetryfoundation.org` categories in our dataset.

dataset is going to contain concepts and uses of these categories that are not aligned with more historical sources of poetry.

TOPIC	METADATA	EVENT	MOOD	OTHER
famous love poem	spanish poem	anniversary poem	encouragment poem	poets
poem for kids	haiku poem	preschool graduation poem	silly poem about seasons	roses are red poem
funny poem	langston hughes poem	fathers day poem	nasty poem	analysis of the poem on the pulse of the morning
life poem	the pearl poem	8th grade graduation poem	heartbroken poem	famous poets
vampire poem	dr.seuss poem	-mothers day poem	envy poem	poem and quotes
romantic love poem	name poem	funeral poem	intimate poem	why do you hate me daddy poem
poem about friends	shape poem	easter poem	sweet poem	types of poem
country poem	italian poem	obituary poem	cry poem	dickinson poem explanations
poem about snowman	poem for my fathers	mom poem	suicidal poem	you dont love me but her poem
mothers daughter poem	famous friendship poem	graduation poem	funny love poem	when god created mothers poem
best friend poem	limerick poem	mothers day poem by helen steiner rice	emo poem	poets and writers
concrete poem	the highwayman poet	free christian mothers day poem	sad love poem	how to write a poem
sexy poem	handprint poem	baby shower gift thank you poem	inspirational stories poem	academy of american poets
cheating poem	short poem	poem about birthdays	inspirational poem	poem
i love you poem	famous love poem	free preschool graduation poem	gangster love poem	walking through the fires of the shadows of the dead poem
multicultural poem	ee cummings poem	birthday invitation poem	funny mothers day poem	
a poem for my daughter	poem rainbow bridge	birthday poem	inspirational mothers day poem	
sex poem	footprints in the sand poem	mothers days poem	sad missing you death poem	
trail of tears poem	robert frost poem	mothers day cards and poem	hurt poem	
kids poem	maya angelou poem	free retirement poem	suicide poem	

**Table 5.3:** 20 example queries classified into the broad categories of TOPIC, METADATA, EVENT, MOOD, and OTHER. Note that some queries have multiple labels; the frequency of each category is given in Table 5.2

Category	Number of Poems $N$
Islam	19
Buddhism	20
Other Religions	41
Infancy	44
First Love	52
Horror	60
Birth	63
Birthdays	63
Fairy-tales	64
Legends	64
Film	66
Photography	66
Gardening	69
Judaism	69
Architecture	71
Design	71
Indoor Activities	71
Dance	78
Theater	78
Country Life	82
Town	82
Midlife	86
Ghosts	88
the Supernatural	88
Divorce	113
... 99 categories with $100 < N < 1000$ ...	
Death	1035
Religion	1049
History	1067
Politics	1067
Activities	1461
Love	1608
Arts	2176
Sciences	2270
Nature	2644
Relationships	3050
Social Commentaries	3050
Living	3946

**Table 5.4:** Frequency ordered categories present in `poetryfoundation.org` dataset, categories in the middle are elided only those with frequencies lower than 100 and higher than 1000 are present in this list.

## 5.2 Poetry Retrieval Models

In this section, we discuss the query-expansion models and their semi-supervised combination that we use to pool documents for labeling. We also present models based on `poetryfoundation.org` categories and an existing emotion-word association dataset.

### 5.2.1 Query Expansion Models

In information retrieval, models that submit a query and improve their results automatically from the documents retrieved at the highest ranks are said to be pseudo-relevance feedback models (PRF models). Relevance modeling is a probabilistic model for performing this pseudo-relevance feedback (Lavrenko and Croft, 2001), where the top-ranked documents under an initial query are used as a model of relevance. Using this model, more query terms can then be sampled from this count-based probability distribution. When a corpus other than the target corpus is used for expansion, these techniques are called external expansion (Diaz and Metzler, 2006). We consider both traditional relevance feedback and external expansion.

For our external collection, we used the version of Wikipedia distributed with the 2018 TREC News Track<sup>1</sup>. Wikipedia and other knowledge bases have been shown to be useful for query-expansion models (Diaz and Metzler, 2006; Dalton et al., 2014; Xiong and Callan, 2015).

Unfortunately, these models have two hyperparameters:  $d$ , the number of top pseudo-relevant documents to consider, and  $k$  the number of terms to sample from the relevance model. In order to select reasonable values for these parameters, researchers typically use some held-out queries or datasets. Instead of trying to select perfect values, we select some reasonable parameter settings that work well on other data and consider these as individual models.

---

<sup>1</sup><https://ir.nist.gov/all-enwiki-20170820.tar.xz>

We consider  $d \in \{3, 10, 30\}$  feedback documents and expanded queries of size  $k \in \{10, 50, 100\}$  additional terms. By varying the hyper-parameters of this model for both Wikipedia feedback and Poetry-based feedback source  $\in \{\text{wiki}, \text{poetry}\}$ , we generate 18 different models. Another model we consider is the original query-likelihood ranking (Ponton and Croft, 1998; Zhai and Lafferty, 2001; Croft et al., 2010) used to select the pseudo-relevant documents. It is useful to include this as a baseline because sometimes expansion models can actually weaken the original ranking if they introduce too much concept drift.

### 5.2.2 Semi-Supervised Expansion Model Combination

Using the categories from the `poetryfoundation.org` dataset, we imagine that they are queries, and the documents labeled with these categories the relevance data. We evaluate our 19 models on this artificial, semi-supervised dataset in order to learn a single combined model, as a way of exploring the effect of selecting better hyperparameters.

We pool our 19 models expansion models to a depth of 200 and generate input files for Ranklib, where each model is encoded as a feature (Dang, 2015). Using the provided coordinate ascent algorithm, we are able to learn a linear combination of these features based on the category labels available in the `poetryfoundation.org` dataset. We train models in a cross-fold cross-validation setting and construct a baseline system for our extracted poetry dataset from the average of these models.

### 5.2.3 Emotion Vector Model

Poetry is often written to convey emotion rather than content. In order to explore whether we can improve retrieval with this insight, we obtained an emotion lexicon (Mohammad and Turney, 2013). While sentiment has been used for information retrieval before (Chelaru et al., 2013; Vural et al., 2014; Zhang, 2015; Wakamiya et al., 2015), including for diversification (Aktolga, 2014), we are interested in more than just positive or negative sentiment and had to look outside standard datasets.

Mohammad and Turney developed the NRC Sentiment dataset which contains positive and negative sentiment as well as scores for 8 emotions: *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise*, *trust*. Their lexicon contains 14,182 words that are labeled with these senses.

We compile their lexicon into a vector of floating point values for each word,  $w$ :  $\vec{E}_w$ .

We transform each poem into a probability distribution over words where  $\text{tf}(w, D)$  is the count of the word  $w$  in the document  $D$  and  $|D|$  is the count of all words in the document.

$$P(w|D) = \frac{\text{tf}(w, D)}{|D|}$$

Given this probability distribution, we can accumulate emotion vectors for each word in a document proportional to the occurrence of each word, in a normalized manner.

$$\vec{E}_D = \sum_{w \in D} P(w|D) \cdot \vec{E}_w$$

The resulting document vectors:  $\vec{E}_D$  have a score for each of the 8 emotion dimensions, and are available for any document.

We now wish to acquire vectors for the queries in our dataset. Since we have vectors with floating point interpretations, we cannot use the count-based interpretation of Relevance Model for feedback, as we did in Section 5.2.1. Instead, we look to the Rocchio model (Rocchio, 1971), which is based upon the centroid of document vectors retrieved in response to a query.

Given a query  $Q$  and the vectors for the documents which are most highly-ranked  $\vec{E}_Q(d) = \{\vec{E}_{Q,0}, \vec{E}_{Q,1} \dots \vec{E}_{Q,d-1}\}$ , we are able to construct a query vector as the sum of these  $E_Q(k)$  vectors for a given ranking depth  $d$ .

$$\vec{E}_Q(k) = \sum_{i=0}^{k-1} \vec{E}_{Q,i}$$

This model lacks the hyperparameter  $k$ , which is the number of terms used for expansion, because the queries and documents are dense vectors and and it is reasonable to represents

all dimensions. We call this our vector-based pseudo-relevance feedback model “vector PRF” and will use it in our next section with vectors computed in a different manner.

The score of a document under this new query is the dot-product of the document vector  $\vec{E}_D$  with the expanded query vector  $\vec{E}_Q$ .

#### 5.2.4 Categorical Vector Model

We take the `poetryfoundation.org` category set and train a language model for each category based on the poems that were tagged with that category. We leverage our target corpus of poetry as a background model, and perform log-odds naïve bayes classification.

That is, for every term in a document, we estimate the probability that it was drawn from a particular category’s model. We can also estimate the probability that it was drawn from a random poem (called a background model). As is common in text classification tasks, we can then assign a score of the likelihood of it being drawn from a category in comparison to a background model.

To estimate these probabilities, we model each poem as a bag of words. We make a term independence assumption and model the probability of a poem  $D$  being drawn from a model  $M$  as the product of the probabilities of each of its words being drawn from that model:

$$P(D|M) = \prod_{w \in D} P(w|M) \tag{5.1}$$

The probability of a word being drawn from a model is as follows, where  $\text{tf}(w, M)$  is the count of all times the word  $w$  occurs in any document used to estimate  $M$  and  $|M|$  is the count of all words in all documents used for estimation. For a category model, this means we consider all the words in all the documents with that category label, and for the background model, this means all words in all documents, regardless of category.

$$P(w|M) = \frac{\text{tf}(w, M)}{|M|}$$

Since category models are somewhat sparse, we may have zero probabilities. This is a problem for our document scoring (Eq. 5.1) because a zero for any term will cause the whole product to be zero and lose all information estimated.

This is typically solved using some kind of interpolation (Croft et al., 2010), and we choose linear smoothing, setting a hyperparameter  $\lambda \in [0, 1.0)$  to mix in a nonzero background probability that will avoid setting our whole  $P(D|M)$  to zero for a single missing term. Therefore we can generate a  $M_i$  for all 126 categories in our `poetryfoundation.org` dataset and ensure that we never draw a zero probability using our general, background model  $G$ .

$$P_{\text{smoothed}}(w|M_i) = \lambda P(w|M_i) + (1 - \lambda)P(w|G)$$

We are able to create a vector of probabilities based on the categories provided in the `poetryfoundation.org` dataset through these smoothed  $M_i$  language models. Note that since we used these categories to generate our query dataset, we excluded as dimensions those categories also used as queries.

We then follow a vector-based PRF approach for document ranking as described for our emotion vectors in the previous section (§5.2.3).

### 5.2.5 Result Pooling

Using the 19 expansion models, including the original ranking (§5.2.1), the semi-supervised combination model (§5.2.2), and our two vector-based models (§5.2.3,5.2.4) we have the ability to generate 22 ranked lists for each query.

In order to evaluate which models are most effective, we will collect the top results from all of these ranked lists and label them as being actually relevant or not in the construction of our dataset, which we will detail in the next section.



### 5.3 Poetry 20 Query Dataset

Given a sampling of the most frequent poem queries from our logs and a random sampling of `poetryfoundation.org` categories, we generated twenty queries to use for the creation of a full evaluation dataset. We attempted to use all of the popular queries from our query log analysis and then randomly filled from the `poetryfoundation.org` categories.

We take a the set of 22 models described in the previous section and pool the top-10 results from each. Over twenty queries, this gives us 1347 query-document pairs to label at this depth.

This dataset is the first true retrieval dataset ever built on poetry tasks and data. After selecting our queries, we performed judgments on a subset to get a feel for the task, and then wrote longer descriptions in an attempt to remove ambiguity from the labeling process (Table 5.6). Our descriptions are somewhat arbitrary, but are more precise than the single terms selected as queries. For event-based queries, we tried to make it clear that this should not be the annotators’ personal opinion, e.g., “How suitable is this poem for someone’s graduation?”

	Measure	<code>poetryfoundation.org</code>	Poetry 500k
Disk Size	Gzipped JSON	8.6MiB	679 MiB
	Inverted & Direct Index	23MiB	1104 MiB
Corpus Stats	Total Poem-Pages	12,959	847,985
	Unique Poem-Pages	12,959	598,333
	Total Terms	3,385,372	179,198,030
	Average Poem Length	211.3	268.5
Dataset Stats	Categories	136	0
	Queries Collected	0	20
	Collection Method	Curated	Automatic

**Table 5.5:** Statistics for the two retrieval corpora: `poetryfoundation.org` and our own extracted collection. On disk sizes are calculated with `du -h`, we did not actually search for duplicates in the `poetryfoundation.org`.

Unfortunately, although the query log displayed a large number of multiterm queries, there was usually only one term that wasn’t attempting to identify genre (Table 5.1) and the

poetry categories were similarly single-term (Figure 5.1). We suspect that given access to a poetry search engine, users may be more inclined to generate more specific search terms and longer queries without fear of concept drift. However, this means that we are not able to explore term dependency models with this dataset.

Since our queries are so short, our focus on query expansion models is justified. Consider the “photography” query: poems about photography may discuss film, cameras, lenses, shutters, composition and a whole host of other words that would make it obvious to a human that it is about photography. Submitting a single term query is going to drastically limit recall compared to expansion.

Query	Source	Description
satire	Category	How relevant is this poem to a search for satirical poetry?
photography	Category	How relevant is photography to this poem?
buddhism	Category	How relevant is buddhism to this poem?
dance	Category	How relevant is dance or dancing to this poem?
death	Category	How relevant is death to this poem?
ancestor	Category	How relevant is the concept of ancestors or ancestry to this poem?
ethnicity	Category	Does this poem discuss ethnicity or identity?
relationships	Category	Does this poem discuss relationships, either romantic or not?
doubt	Category	How relevant is the concept of doubt to this poem?
sorrow	Category	Does this poem describe sorrow or is it deeply sad in some way?
flowers	Category	Is this poem about or filled with flowers?
mother’s day	QLog	How suitable is this poem for Mother’s Day? Does it describe mother-child relationships?
graduation	QLog	How suitable is this poem for someone’s graduation?
love	QLog	How relevant is love to this poem?
friendship	QLog	How relevant is friendship to this poem?
easter	QLog	How suitable is this poem for Easter? Does it describe easter holidays?
inspiration	QLog	Would this poem be inspirational to someone?
teacher	QLog	Does this poem describe a teacher-student relationship or something about teaching?
funny	QLog	Would this poem be funny to someone?
wedding	QLog	Would this poem be suitable for a wedding, or does it describe marriage in some way?

**Table 5.6:** Queries and Descriptions used for Poetry 20 Dataset

### 5.3.1 Crowdsourced Label Collection

While crowdsourcing does not necessarily generate the highest quality labels, the relative cost has made it the defacto method for collection of judgments in recent years (e.g., Bailey et al. (2016)) and for at least some recent TREC tasks the quality of crowdsourced judgments has had no overall effect on the ranking of systems (Dietz et al., 2017b; Allan et al., 2017).

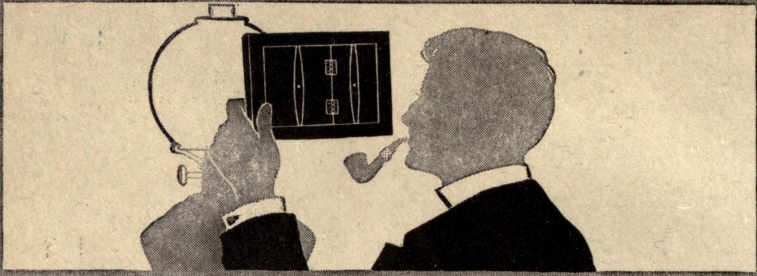
We collected labels of Not Poetry (-1), Not-Relevant (0), Maybe-Relevant (1), and Definitely-Relevant (2). Annotators were given the query and our description as presented in Table 5.6. Statistics are available about the crowdsourced labels in Table 5.7.

This was a challenging task for annotators. Although we had multi-level judgments of “Maybe-Relevant” and “Definitely-Relevant”, our annotators struggled with this distinction: only achieving 32.7% agreement with those levels kept separate. Agreement shoots up to 60.8% if we collapse the distinction between maybe and definitely relevant. This suggests that annotators had a difficult time telling the quality of relevance of results. Additionally, with the *general retrieval quality* being so high – 1081 of the 1347 pages had someone label them as relevant – most documents reviewed were actually poems and were actually on topic, at least according to one of two annotators.

The queries with the lowest agreement, specifically “photography”, and “graduation”, were the best at pulling up false-positives from our poetry classifier based on visual features: there were many advertisements for Kodak film for the former query (e.g., Figure 5.2) and the query “graduation” tended to pull up author bios from the back of a book (e.g., Figure 5.3), which might have centered text and appear poetic to our system (§3.2.1).

Query	Count	Relevant	Not-Poetry	Agreement	M-Agreement
satire	69	39–61	2	0.609	0.275
photography	65	15–37	24	0.338	0.231
buddhism	54	26–45	5	0.556	0.259
dance	53	27–47	4	0.528	0.226
mothers day	74	39–60	7	0.622	0.351
graduation	62	18–44	12	0.435	0.290
death	51	37–49	1	0.745	0.529
inspiration	78	47–72	3	0.628	0.346
teacher	54	26–41	12	0.537	0.370
ancestors	64	42–59	4	0.672	0.406
ethnicity	51	22–41	4	0.510	0.235
funny	60	34–54	1	0.633	0.333
relationships	72	28–47	21	0.514	0.292
wedding	65	43–61	3	0.662	0.292
love	67	38–58	2	0.672	0.299
doubt	64	40–60	4	0.625	0.234
flowers	73	44–63	7	0.671	0.356
friendship	60	36–54	3	0.633	0.317
sorrow	69	50–65	1	0.725	0.449
easter	66	52–63	2	0.803	0.424
Overall:	1347	703–1081	122	0.608	0.327

**Table 5.7:** Agreement and Relevance information for our Poetry 20-Query Dataset



Prints by Gaslight

We are as eager to have you  
make good prints as you are—  
that's the reason for

# VELOX

a photographic paper that *fits*.

*Use the new Contrast Velox with flat negatives.*

*At your dealer's.*

NEPERA DIVISION,  
EASTMAN KODAK CO., ROCHESTER, N. Y.

Please mention Practical Photography when writing Advertisers

**Figure 5.2:** A Kodak advertisement in a book on photography (Fraprie, 1915); this was identified as poetry by our algorithm, and is a false-positive search result for “photography”. This book contains 9 full pages of advertisements at the end of its content.

Graduated from the University of California at Berkeley with honors in 1989, with a B.A. in Rhetoric.

In 1986 was hired as a student employee with the Regional Oral History Office to transcribe oral histories. Has stayed on since graduation and is currently an editorial assistant.

Writes short stories, and is completing her first novel about life in rural northern California.

**Figure 5.3:** An author biography at the end of a book containing an interview with a local publishing company done by a University Library (Rather et al., 1994). This is another false positive identified by our algorithm, and comes up as a result for “graduation”.

## 5.4 Retrieval Model Evaluation

Using our novel poetry dataset, we are able to evaluate the features and models that we used to collect judgments. Since the prior probability of relevance in this dataset is so high, distinctions between features are currently not significant in most comparisons.

### 5.4.1 Experimental Setup

We focus on the minimum relevance judgments in most analysis in order to have more discriminative power; the idea being that we are recovering multilabel judgments by identifying those that more than one annotator viewed as relevant as the true relevant ones and trying to push those up the ranking. That is, we only consider a document to be relevant if all annotators agreed on its relevance.

We look at three retrieval measures.

**mAP** or mean average precision is the mean of the precision at the recall points. It is a traditional measure that emphasizes both precision and recall of the ranking.

**P10** is precision at a depth of 10. Because we pooled our documents to be judged to this depth, this is the measure for which we have complete judgments. Unfortunately, it is rather sparse; with only 11 possible values per query.

**R-Prec** is precision at the depth of the number of known relevant documents for each query.

A perfect system would achieve R-Prec of 1.0 regardless of the number of relevant documents available for each query.

### 5.4.2 Results

Performance for all of our our retrieval models is presented in Table 5.8. Our performance is dominated by expansion models, although query likelihood, a unigram model, is not too far behind because of the high prior-probability of relevance. In general, we see that more documents are better for expansion, and that from Wikipedia, fewer terms seem to work best (Wiki-X), but more terms can work better for expansion from the Poetry corpus itself

(Poems-X). Our Semi-Supervised learning to rank model does not perform well, given that it was tuned on a very different dataset (the `poetryfoundation.org` categories), this is not too surprising.

Our vector-based approaches arrive in dead-last, despite performing feedback atop the QL model, which does decently well itself. This suggests, once more, that term-based classification approaches (from which we constructed the vectors) struggle with poetry. We are not yet succeeding at transferring knowledge from existing poetry collections (however, the `poetryfoundation.org` labels are quite sparse and are far more modern), or from research into emotion word associations (Mohammad and Turney, 2013).

Perhaps, however, the weakness of our vector-based approaches reflects a more general weakness of such approaches: global representations of word meanings can be fragile, and word embeddings built from query context tend to be much better than those built from global context (Diaz et al., 2016; Zamani and Croft, 2016, 2017). More research is needed on a larger poetry retrieval dataset to keep exploring topical and emotional retrieval models.

### 5.4.3 Emotion and Category Vector Performance

Since our category and emotion vector models performed rather poorly, we observe the most differences here. The topical model is far more effective than the emotional model, but both are still quite simple. We present per-query results in Table 5.9. Only for the queries “flowers” and “buddhism” do we find that the emotion vector approach beats the category vectors. This is surprising because both queries are categories in the `poetryfoundation.org` dataset, though they are kept held-out in the category vector representation. Other queries for which we expected emotion to be helpful, e.g., “love”, “death”, and “doubt” are particularly bad with this method of ranking.

We note that the wide variety of topics covered in the `poetryfoundation.org` categories mean that the category vector approach and the emotion vector approaches are not necessarily opposites; there are many emotional and mood words represented in the categories (Figure 5.1)



Model	Parameters	mAP	P10	R-Prec
Wiki-X	$d = 30, k = 10$	0.300	0.600	0.370
Poems-X	$d = 30, k = 50$	0.288	0.635	0.371
Poems-X	$d = 30, k = 100$	0.287	0.640	0.374
Poems-X	$d = 10, k = 10$	0.287	0.635	0.362
Poems-X	$d = 10, k = 100$	0.285	0.635	0.370
Wiki-X	$d = 10, k = 10$	0.284	0.580	0.368
Poems-X	$d = 30, k = 10$	0.284	0.640	0.351
QL	N/A	0.283	0.625	0.372
Wiki-X	$d = 30, k = 50$	0.283	0.580	0.361
Poems-X	$d = 10, k = 50$	0.282	0.630	0.361
Wiki-X	$d = 10, k = 50$	0.279	0.555	0.371
Wiki-X	$d = 30, k = 100$	0.274	0.520	0.367
Wiki-X	$d = 10, k = 100$	0.273	0.540	0.363
Semi-Supervised	N/A	0.271	0.615	0.361
Wiki-X	$d = 3, k = 10$	0.266	0.515	0.343
Wiki-X	$d = 3, k = 50$	0.259	0.545	0.345
Poems-X	$d = 3, k = 10$	0.256	0.595	0.315
Wiki-X	$d = 3, k = 100$	0.252	0.510	0.335
Poems-X	$d = 3, k = 50$	0.246	0.600	0.302
Poems-X	$d = 3, k = 100$	0.238	0.595	0.289
Category Vectors	$\lambda = 0.9$	0.162	0.440	0.233
Category Vectors	$\lambda = 0.8$	0.161	0.440	0.228
Category Vectors	$\lambda = 0.7$	0.159	0.440	0.229
Category Vectors	$\lambda = 0.6$	0.159	0.440	0.222
Category Vectors	$\lambda = 0.5$	0.159	0.440	0.225
Category Vectors	$\lambda = 0.4$	0.159	0.435	0.222
Category Vectors	$\lambda = 0.3$	0.158	0.435	0.223
Category Vectors	$\lambda = 0.2$	0.152	0.420	0.211
Category Vectors	$\lambda = 0.1$	0.152	0.450	0.202
Emotion Vector	N/A	0.098	0.145	0.132

**Table 5.8:** Performance, ordered by mAP, of various retrieval models on our Poetry dataset.

Query	Category			Emotion		
	mAP	P10	R-Prec	mAP	P10	R-Prec
graduation	0.270	0.3	0.462	0.169	0.2	0.308
ancestors	0.242	0.9	0.238	0.073	0.1	0.048
dance	0.209	0.4	0.294	0.086	0.1	0.029
ethnicity	0.208	0.3	0.364	0.202	0.3	0.409
satire	0.203	0.4	0.286	0.156	0.2	0.245
easter	0.193	0.7	0.192	0.121	0.0	0.192
mothers day	0.171	0.4	0.244	0.068	0.0	0.089
doubt	0.170	0.6	0.250	0.026	0.0	0.000
love	0.169	0.6	0.210	0.037	0.0	0.000
death	0.167	0.3	0.288	0.108	0.2	0.115
buddhism	0.166	0.3	0.219	0.230	0.5	0.281
sorrow	0.163	0.5	0.260	0.091	0.4	0.100
friendship	0.154	0.6	0.250	0.050	0.1	0.056
teacher	0.148	0.4	0.231	0.050	0.1	0.038
wedding	0.147	0.6	0.233	0.057	0.0	0.070
funny	0.129	0.5	0.176	0.099	0.1	0.118
relationships	0.123	0.3	0.143	0.067	0.1	0.071
flowers	0.114	0.3	0.136	0.177	0.4	0.295
inspiration	0.089	0.4	0.128	0.059	0.1	0.106
photography	0.014	0.0	0.059	0.024	0.0	0.059

**Table 5.9:** Per-Query Performance of Category and Emotion vectors, ordered by mAP of the Category vector approach.

## 5.5 Remaining Challenges for Poetry Retrieval

One of the open challenges of our new dataset is the quantity of queries which appear easy from an IR perspective. For instance, the query “doubt” had a pool of 64 results. At least one annotator for each page considered the poetry on that page relevant to the initial query, provided it was actually poetry. This means that this query is not useful at all for comparative ranking of systems with the current set of relevance judgments at the current evaluation depth.

Although one of the challenges of poetry retrieval is that a love poem does not necessarily mention the word “love”, we have the largest searchable and public collection of poetry in the world. Therefore, in a collection of 600,000 pages with poetry, there are sufficiently many poems mentioning topical words that we do not actually observe the vocabulary mismatch problem under traditional IR pooling at this depth. Since annotators consistently showed such strong disagreement over the quality of relevance (see the M-Agreement column in Table 5.7) we suspect it will be difficult to collect assessments of the relative quality of different rankings.

Future work will need to explore alternate methods of document collection building, including stratified sampling and working with larger pools. In order to improve relative relevance understanding (since the prior probabilities for common poetry queries are so high) we also plan to explore pairwise relevance judgments, where annotators are asked to choose which poem better represents the query. Even if most results are relevant, having a stronger ordering may allow us to better rank systems.

Additionally, now that this poetry is available and a real system can be built, users of a pure-poetry retrieval system may issue more complicated queries. With more complicated queries, the high prior probability of relevance may be less of an issue for evaluation.

### 5.5.1 Error Analysis and Deeper Results

In this section we take a look at some examples retrieved for queries based on our deep dive into vector performance. The best category vector query was “graduation”, and the worst was “photography”. The best emotion vector query was “buddhism” and the worst was also “photography”, and the second worst was “doubt”. Results are taken from Table 5.9. We explore results from the unigram query-likelihood model ranking.

#### 5.5.1.1 Photography Results

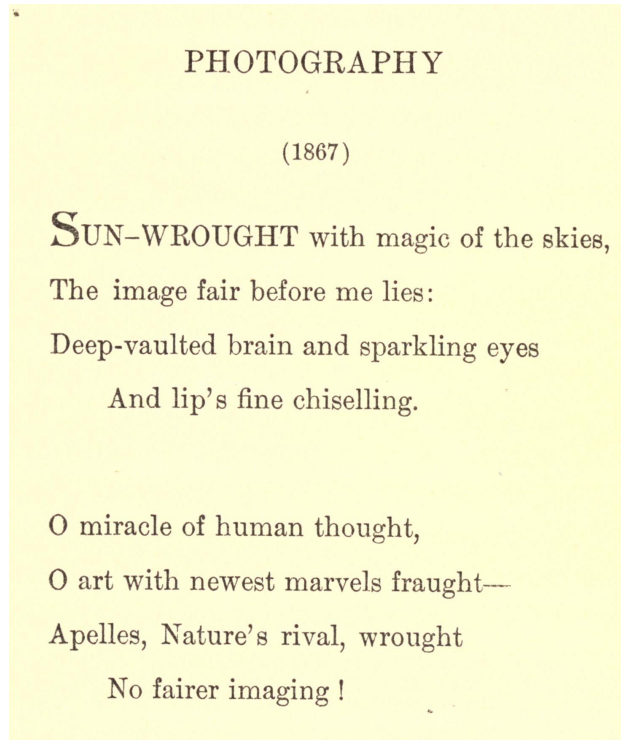
Taking a deeper look at the photography results, we find that in addition to non-poetry false-positives in terms of advertisements (rank 1 & 3) (e.g., Figure 5.2), and scientific text at the end of a chapter (rank 4), (e.g., Figure 5.5). We find some poetry that is not relevant as well: (rank 6) Figure 5.6 sees the realism of photography used as a metaphor to discuss (and argue against) a trend in fiction toward realism. We do find true positives, e.g., at rank 2 there is a poem about photography and its invention (Figure 5.4), but there we are saved by the presentation of the poem alongside its title, as photography is not mentioned in the two stanzas of contents.

#### 5.5.1.2 Graduation Results

Graduation was one of our most successful queries, by most models. Although there were false positives, as previously discussed (such as author biographies, e.g., Figure 5.3), there is a large amount of poetry written about graduation and the emotions thereof, of family (Rank 1: Figure 5.7), personal experiences (Rank 3: Figure 5.8), and misspent youth failing to take good advice (Rank 19: Figure 5.9). Perhaps the last could be marked as slightly less relevant, since it is primarily about advice and only secondarily about graduation.

#### 5.5.1.3 Doubt Results

One of the interesting things that identifies highly-ranked poetry about doubt is the prevalence of religious poetry about doubt. About 10 of the top 20 results are works of poetry



**Figure 5.4:** A poem about the invention of photography that references Apelles, an ancient Greek painter. Note that if this poem were presented without the title, we would be unable to tell that it was describing poetry without significantly more effort.

that discuss doubt in the context of Christianity (e.g., Figure 5.11). A significant portion of the rest of the results (7) actually refer to doubt in a romantic context; where the opposite is fidelity. The top result is from a “calendar” built out of Shakespearean quotes (Figure 5.10). The poem at rank 20 cites a fair amount of religious entities, e.g., “Hell” and “Death” and personifies “Doubt” (Figure 5.12).

iron, steel, and cement will follow with an opportunity to branch out into electro-plating, photography, or special branches which students may have an opportunity to enter.

Agricultural chemistry is given by experimental lectures, recitation, and laboratory work including the following subjects: Ingredients of Plants, Food Requirements of Plants and their Sources, Soil Types—Potato and Truck Soils, Fruit Soils, etc., Chemical Composition of Soils, Soil Exhaustion and Conservation, Methods of Determining Needs of Soils, Farm Manures, Commercial Fertilizers and their Rational Use.

#### PHYSICAL TRAINING.

See regular course in physical training page 14.

**Figure 5.5:** A mention of the word “photography” that appears in the early ranks of our poetry search; this document does not contain poetry but mentions the topic. This is much more common for “photography” than other queries.

#### SIR WALTER

THE Critic says Romance is dead, and we  
Must cease to read her tales of ancient war,  
That only children love fair fancy's lore,  
For “fiction must reflect reality.”  
But art is other than anatomy :  
Romance is true as science is—nay, more ;  
Its fair mirage portrays a haunted shore,  
Outside the field of “pen photography.”

Therefore, I send this hail, O Scott, to thee  
And to the sturdy children of thy pen,  
Peasant and peer, true women and brave men—  
Bold Quentin Durward and sweet Isabel,  
Meg Merrilies and old Mortality,  
Di Vernon and the Knight of Avenel.

**Figure 5.6:** A mention of the word “photography” in a poem that is not actually about photography, but using it as a metaphor in the phrase “pen photography” to describe realistic fiction.



### KITTY'S GRADUATION.

DUBLIN Alley jisht was crazy, jubilation was the rule,  
Chewsday week whin Kitty Casey won the honors at the  
school.

Shure, the neighbors had been waitin', all impatient of delay,  
For to see her graduatin' on that most important day.

Eddication is a power, an' we owned wid one accord  
Casey's girl's the sweetest flower ever blossomed in the  
ward,

Whin, wid dress white as the daisy, but wid cheeks that  
shamed the rose,

We beheld wee Kitty Casey in her graduation clo'es.

Now, this Casey loved his daughther in a most indulgent  
way,

An' he spent his gold like wather for her graduation day.  
Sich a dale of great preparin'! Shure, ye'd think she was a  
bride;

Sorra hair was Casey carin' for a blessed thing beside.

For whin Casey once comminces, faith, he niver stops at all,  
An' he dressed her like a princess at a Coronation Ball.

An' 'twas Madame Brigette Tracy for dressmaker that he  
chose,

For to fit out Kitty Casey in her graduation clo'es.

71

**Figure 5.7:** A poem (most likely a song) about a woman named “Kitty Casey” written in dialect about the emotional investment of a family in her graduation, amongst other possible themes.

### ALUMNI

Queer pencilings scribbled in a book—

Class numerals, some mystic signs in Greek—

Poignant the glad-sad memories

Of Graduation Week.

**Figure 5.8:** The last stanza of a poem (continued from the previous page, which does not mention “graduation” about the “poignant glad-sad” experience of Graduating and leaving behind friends and experiences.

## ADVICE

When I was but a little boy,  
My grandad used to say:  
“Learn something that is useful, lad,  
Each hour of the day;  
And when your *head* is filled, you’ll find,  
It’s quite a simple plan,  
To fill your empty pocketbook,  
When you become a man.”

But Youth is proud Experiment,  
And Age, Experience:  
It’s strange, they’re always alien—  
A queer coincidence!  
And so I failed to heed his words,  
As boys are wont to do;  
For I was young, and he was old,  
And life was rosy-hue.

My little seat-mate, Billy Elm,  
Sure loved his thumb-marked books:  
He’d study them with earnest mien,  
And I, my fishing hooks;  
But when our graduation came,  
He led the honor roll,  
While I disported in the shade  
Of Huckin’s swimming hole.

118

**Figure 5.9:** A poem about coming of age and being disappointed with ones’ achievements when the time for graduation has come after ignoring advice from ones’ elders.



MARCH 2nd

**D**OUBT thou the stars are fire ;  
Doubt that the sun doth move ;  
Doubt truth to be a liar ;  
But never doubt I love.

*Shakespeare.*

**Figure 5.10:** A quote from a Shakespearean sonnet about “doubt” and “love”, which is highly ranked for both queries due to the length of the document, but at rank 1 for “doubt”. The book contains a quote for each day of the year.

III.

The cloud that filled my night was doubt ;  
The night of doubt was black with me ;  
There was no dawning, seemingly,  
Until her star came shyly out—

Came out between the shades that fell  
Athwart my pathway, blindly trod ;  
Came like a gleam of joy from God,  
To be about me like a spell.

My doubt was not a doubt of love,  
Nor doubt of goodness undefined,  
Nor disbelief in human kind,  
Nor doubt of Him who rules above.

It was the doubt of self which hung  
Before me like a misty veil ;  
To me appeared no Holy Grail ;  
There was no guide to which I clung.

I wandered lonely, blindly led,  
As one may wander in a dream,  
While knowing there is no supreme  
And living way thereon to tread.

5

**Figure 5.11:** A Christian poem about religious doubt and personal tribulations. Religion is a very common context for “doubt” in our collection of poetry.

THE PICKERING MS.

The Babe that weeps the Rod beneath  
Writes Revenge in realms of Death.  
He who mocks the Infant's Faith  
Shall be mock'd in Age and Death.  
He who shall teach the Child to doubt  
The rotting Grave shall ne'er get out.  
He who respects the Infant's Faith  
Triumphs over Hell and Death.  
The Child's Toys and the Old Man's Reasons  
Are the Fruits of the Two Seasons.  
The Questioner, who sits so sly,  
Shall never know how to reply.  
He who replies to words of Doubt  
Doth put the Light of Knowledge out.  
A Riddle, or the Cricket's cry,  
Is to Doubt a fit Reply.  
The Emmet's Inch and Eagle's Mile  
Make lame Philosophy to smile.  
He who doubts from what he sees  
Will ne'er believe, do what you please.  
If the Sun and Moon should doubt,  
They'd immediately go out.

The Prince's Robes and Beggar's Rags  
Are Toadstools on the Miser's Bags.  
The Beggar's Rags, fluttering in air,

211

**Figure 5.12:** A poem that is clearly influenced by religion and directly personifies many concepts, including “Doubt”. This poem is highly ranked in a unigram search for “doubt” but is not straightforward in interpretation.

## 5.6 Discussion

Often, information retrieval research focuses on test collections that have content value, where sources are solely informative. Collections of music, commercial products, and books are typically considered to be the in the domain of recommender systems, where the dominant approach is to leverage user behavior to learn similarities between items and user profiles for suggestion. In this cold-start domain, we have identified some key research questions: how to collect labels of subjective utility from annotators, how to explore recall in a meaningful way in rich collections, and how to incorporate emotional meaning into a retrieval system.

We hope that in future efforts that we can better understand the information needs of users and researchers in such datasets and develop techniques for improving access to such rich collections where vocabulary mismatch is a difficult problem to evaluate due to subjectivity.

## CHAPTER 6

### CONCLUSION

In this work, we have developed novel datasets for three tasks centered around poetry extracted from digitally scanned books.

First, we defined poetry identification as a task, where our goal is to locate poetry within larger works. In doing so, we found that content-based approaches generalize poorly to novel data and that visual features are currently the most robust to the variety present in this medium.

Using our state-of-the-art poetry identification models, we automatically extract and de-duplicate a large dataset of 600,000 pages that are identified as poetry, with very high accuracy. Our approaches are highly scalable and can be straight-forwardly applied to the millions of digitally scanned books that are publicly available through various libraries.

Next, we look at named entity recognition within poetry. Dealing with the challenges of our domain leads us to a simpler, more straightforward neural architecture that skips over traditional pre-processing steps. We study labeling curves, learning curves and the features that are necessary for our model’s performance on our new NER dataset. We find that multitask learning with existing NER datasets on news works quite well and suggests that poetry NER may hold value for the NLP community as a task.

Finally, we collected a judged set of queries to evaluate an information retrieval task over poetry data based on real user data in query logs and online poetry categories. We present lessons learned from the label collection process and provide a small comparison of reasonable methods for this task, despite a high prior probability of relevance.

With our contributions in terms of datasets and analysis, poetry can be a useful resource for better understanding models and tasks that are core to the IR and NLP communities. Additionally, our related work is full of researchers working on small, private collections of poetry and our large public datasets can have a large impact on the reproducibility and effectiveness of similar studies in the future.

## 6.1 Contributions

In chapter 3, we introduce and develop a poetry identification task alongside a novel dataset of 2,814 labeled pages from 1,381 digitally scanned books. These pages were collected with a mix of active-learning approaches in order to expose any problems with generalization and to ultimately collect a large collection of poetry. We showed that content models, particularly neural models show promise, but fail to generalize well to new books.

We devise an efficient model based on random forests over formatting features, and execute it over 50,000 books. After de-duplication, we find that there are 600,000 unique pages with poetry in this collection which had 17 million regular pages. In the future, we plan to run this on larger collections of books: early results suggest that a 250,000 books have about 3 million pages worth of poetry, and online libraries have tens of millions of scanned books now. With larger collections of poetry collected in this manner, we can train more robust content models and explore the challenge of generalization.

In chapter 4, we explore named entity recognition (NER) on poetry. We deeply analyze the needs of poetry-based entity recognition. In future studies, we hope to look closer at the use of personified concepts as entities as well as linking the entities found in poetry to knowledge bases for better understanding and representation of poems. We evaluate the different features of a modern neural NER model on poetry data, and find that cross-training on existing News NER datasets is the only critical feature. We don't find our social media dataset (WNUT-16) to be as helpful, but this may be due to its smaller size. The success of our news-based cross-training in particular contributes a promising approach for

historical and literary NLP generalization, which has been shown to be particularly difficult on part-of-speech-tagging (Bamman, 2017).

In chapter 5, we explore information retrieval over poetry. Between our query-log study and our use of human-curated poetry labels from the `poetryfoundation.org` dataset, we develop a sense of user information needs from poetry. If we take our poetry data and make a live search system available to experts, we expect to see deeper and richer information needs. Nevertheless, our dataset involves 20 queries and over one thousand document judgments, which annotates 22 models fully to a depth of 10.

## 6.2 Future Work

First and foremost, we plan to extend our poetry identification models to millions of books and collect the world’s largest collection of poetry. Using such a large dataset, we hope to explore content models again, using poetry-specific word embeddings and representation learning, to further improve our identification techniques.

This poetry collection introduces a number of information extraction challenges that we have not yet delved into: e.g., identifying the metadata of this poetry: title, author, original source, meter, etc. Using insights from our NER model, we hope to identify poetry at the word level, and can possibly generate a large amount of ground truth for this from our duplicate poems.

In future work, we hope to explore the effect that poetry data might have on training more generalizable NER models: can we influence performance on news data? Additionally, larger traditional datasets may lead to more performance on poetry data. Exploring the temporal diversity of our works is important for our ability to generalize: Pennacchiotti and Zanzotto (2008) found that performance of modern-trained data degraded with the age of texts analyzed. While we tried to preserve the fairness of our collection on the book level, publication data is often missing or incorrect (Foley and Allan, 2015) and we did not

take publication date into account as a result, but analyzing the metadata could provide interesting information.

One of the clearest next steps for our allusion work is entity linking: where we take the detected named entities and attempt to link them to a knowledge base, such as Wikipedia. Allusions may provide challenge since (unlike in other domains) poetry does not usually explain any context of entities that are mentioned. Other classical sequence tagging problems will be interesting to look at in poetry, but require some deeper linguistic thought: how do parts of speech apply to poetry?

Our retrieval work is fairly preliminary, but some obvious improvements would be to provide faceted search: if a user types in love poems, offer commonly-co-occurring terms as options for refining their search. Making our data and search system available to digital humanists will likely generate large, expert queries that will be exciting to support in both an effectiveness and efficiency context.

One future direction for our retrieval corpus is to consider what it would take to build a recommendation system for poetry. Not only will this require new truth data collected by annotators, this will potentially require knowledge of structure, central theme, and other poetic concepts. Users may find poems interesting if they are humorous, or maybe if they are very serious. It is likely that emotion detection and data will be more useful for this kind of task. It would be interesting to see if co-publication of poetry can be a useful starting dataset for this work: poems published in the same book could be thought of as being selected by a single user (the author) and may even be useful for evaluation if data sparsity can be overcome.

Another computer science domain that might find our dataset interesting is in research on speech synthesis: we do not think current algorithms are capable of reading the variety of poetry meters, languages, and style available in our collection. Making this corpus available as audio files with higher levels of fidelity would make poetry more accessible.

We expect the digital humanities to be most interested in this work. Using our poetry datasets, we can now study the popularity of older poetry (at least as seen through the perspective of publishers) by analyzing the results of our duplicate detection on larger sets. Poetry-specific phenomena can be studied: using our NER system, metaphor and simile could be identified and analyzed across millions of poems. Poems about specific historical events could be found and curated to better understand historical events and human sense-making. For these domains and more, retrieval will be the critical task to make sense of computational analysis and to provide the ability to curate forgotten collections of poetry.



## BIBLIOGRAPHY

- Ahmed, M. A. and Trausan-Matu, S. (2017). Using natural language processing for analyzing arabic poetry rhythm. In *Networking in Education and Research (RoEduNet), 2017 16th RoEduNet Conference*, pages 1–5. IEEE.
- Aktolga, E. (2014). *Integrating Non-Topical Aspects into Information Retrieval*. Ir.
- Allan, J., Harman, D., Kanoulas, E., Li, D., Van Gysel, C., and Vorhees, E. (2017). Trec 2017 common core track overview. TREC.
- Alsharif, O., Alshamaa, D., and Ghneim, N. (2013). Emotion classification in arabic poetry using machine learning. *International Journal of Computer Applications*, 65(16).
- Amati, G. and Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389.
- Amitay, E. and Broder, A. (2008). Introduction to special issue on query log analysis: Technology and ethics. *ACM Trans. Web*, 2(4):18:1–18:2.
- André, P., Teevan, J., and Dumais, S. T. (2009). From x-rays to silly putty via uranus: serendipity and its role in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2033–2036. ACM.
- Armstrong, T. G., Moffat, A., Webber, W., and Zobel, J. (2009). Improvements that don’t add up: ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 601–610. ACM.
- Aslam, J., Ekstrand-Abueg, M., Pavlu, V., Diaz, F., and Sakai, T. (2013). Trec 2013 temporal summarization. In *TREC’13*.
- Bai, L., Guo, J., and Cheng, X. (2011). Query recommendation by modelling the query-flow graph. In *Asia Information Retrieval Symposium*, pages 137–146. Springer.
- Bailey, P., Moffat, A., Scholer, F., and Thomas, P. (2016). Uqv100: A test collection with query variability. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 725–728. ACM.
- Balog, K., Serdyukov, P., and Vries, A. P. d. (2010). Overview of the TREC 2010 entity track. Technical report, DTIC Document.
- Bamman, D. (2017). Natural language processing for the long tail. In *Digital Humanities*.

- Baron, A. and Rayson, P. (2008). Vard2: A tool for dealing with spelling variation in historical corpora. In *Postgraduate conference in corpus linguistics*.
- Barros, L., Rodriguez, P., and Ortigosa, A. (2013). Automatic classification of literature pieces by emotion detection: A study on quevedo’s poetry. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 141–146. IEEE.
- Bates, M. J., Wilde, D. N., and Siegfried, S. (1993). An analysis of search terminology used by humanities scholars: the getty online searching project report number 1. *The Library Quarterly*, 63(1):1–39.
- Becker, W., Wehrmann, J., Cagnini, H. E., and Barros, R. C. (2017). An efficient deep neural architecture for multilingual sentiment analysis in twitter.
- Bridges, J. (2015). Poetryfoundation.org poetry crawl. <https://github.com/jacobbridges/poetry-collection-api/blob/master/archive/poetryfoundation.org-scrape-08-15-2015.json.gz>.
- Broder, A. (2002). A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM.
- Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM.
- Can, E. F., Can, F., Duygulu, P., and Kalpakli, M. (2011). Automatic categorization of ottoman literary texts by poet and time period. In *Computer and Information Sciences II*, pages 51–57. Springer.
- Chaker, J. and Habib, O. (2007). Genre categorization of web pages. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 455–464. IEEE.
- Chang, Y.-C., Chu, C.-H., Chen, C. C., and Hsu, W.-L. (2016). Linguistic template extraction for recognizing reader-emotion. *International Journal of Computational Linguistics and Chinese Language Processing*, 21(1):29–50.
- Chaudhuri, P., Dasgupta, T., Dexter, J. P., and Iyer, K. (2018). A small set of stylometric features differentiates latin prose and verse. *Digital Scholarship in the Humanities*.
- Chelaru, S., Altingovde, I. S., Siersdorfer, S., and Nejdl, W. (2013). Analyzing, detecting, and exploiting sentiment in web queries. *ACM Transactions on the Web (TWEB)*, 8(1):6.
- Chen, Y., Liu, Y., Zhou, K., Wang, M., Zhang, M., and Ma, S. (2015). Does vertical bring more satisfaction?: Predicting search satisfaction in a heterogeneous environment. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1581–1590. ACM.

- Choi, K., Lee, J. H., Hu, X., and Downie, J. S. (2016). Music subject classification based on lyrics and user interpretations. *Proceedings of the Association for Information Science and Technology*, 53(1):1–10.
- Clary, K., Tosch, E., Foley, J., and Jensen, D. (2018). Let’s Play Again: Variability of Deep Reinforcement Learning Agents in Atari Environments. In *Critiquing and Correcting Trends in Machine Learning Workshop at Neural Information Processing Systems*.
- Cohen, D., Jordan, S., and Croft, W. B. (2018). Distributed evaluations: Ending neural point metrics. In *SIGIR; LND4IR*.
- Cormack, G. V. and Grossman, M. R. (2016). Scalability of continuous active learning for reliable high-recall text classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1039–1048. ACM.
- Craswell, N. (2009). *Proceedings of the 2009 workshop on Web Search Click Data*. ACM.
- Crawford, J. (1886). *The Poet Scout: A Book of Song and Story*. Funk and Wagnalls.
- Croft, W. B., Metzler, D., and Strohman, T. (2010). *Search engines: Information retrieval in practice*, volume 283. Addison-Wesley Reading.
- Dalton, J., Dietz, L., and Allan, J. (2014). Entity query feature expansion using knowledge base links. In *SIGIR’14*, pages 365–374. ACM.
- Dang, V. (2015). Ranklib, v.2.5-snapshot. <https://sourceforge.net/p/lemur/wiki/RankLib>.
- De Vries, A. P., Vercoustre, A.-M., Thom, J. A., Craswell, N., and Lalmas, M. (2008). Overview of the inex 2007 entity ranking track. In *Focused Access to XML Documents*, pages 245–251. Springer.
- Dehghani, M., Zamani, H., Severyn, A., Kamps, J., and Croft, W. B. (2017). Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM.
- Demartini, G., Iofciu, T., and De Vries, A. P. (2010). Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation*, pages 254–264. Springer.
- Devlin, J. (2018). Bert github repository readme. <https://github.com/google-research/bert>.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Diaz, F. and Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM.

- Diaz, F., Mitra, B., and Craswell, N. (2016). Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*.
- Díaz-Agudo, B., Gervás, P., and González-Calero, P. A. (2003). Adaptation guided retrieval based on formal concept analysis. In *International Conference on Case-Based Reasoning*, pages 131–145. Springer.
- Dietz, L., Kotov, A., and Meij, E. (2017a). Utilizing knowledge graphs in text-centric information retrieval. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 815–816. ACM.
- Dietz, L., Verma, M., Radlinski, F., and Craswell, N. (2017b). Trec complex answer retrieval overview. TREC.
- Donaldson Jr., S. J. (1860). *Lyrics and other poems*. Lindsay and Blakiston; Philadelphia.
- Explosion AI (2018). <https://spacy.io/>.
- Farajidavar, N., Kolozali, S., and Barnaghi, P. (2017). A deep multi-view learning framework for city event extraction from twitter data streams. *arXiv preprint arXiv:1705.09975*.
- Foley, J. and Allan, J. (2015). Retrieving time from scanned books. In *Proceedings of the 37th European Conference on Information Retrieval*, pages 221–232. Springer.
- Foley, J., O’Connor, B., and Allan, J. (2016). Improving entity ranking for keyword queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2061–2064. ACM.
- Fraprie, F. R. (1915). *How to choose and use a lens*. American Photographic Publishing Co.
- Friedland, L. and Allan, J. (2008). Joke retrieval: recognizing the same joke told differently. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 883–892. ACM.
- Galves, C. and Faria, P. (2010). Tycho Brahe parsed corpus of historical Portuguese. <http://www.tycho.iel.unicamp.br/>.
- Ganguly, D., Leveling, J., and Jones, G. J. (2014). Retrieval of similar chess positions. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 687–696. ACM.
- Ghazvininejad, M., Shi, X., Choi, Y., and Knight, K. (2016). Generating topical poetry. EMNLP, pages 1183–1191.
- Gimpel, K., Schneider, N., O’Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.
- Grossman, M. R., Cormack, G. V., and Roegiest, A. (2017). Automatic and semi-automatic document selection for technology-assisted review. SIGIR, pages 905–908. ACM.
- Guo, Z., Wang, Q., Liu, G., Guo, J., and Lu, Y. (2012). A music retrieval system using melody and lyric. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 343–348. IEEE.
- Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.
- Hamidi, S., Razzazi, F., and Ghaemmaghami, M. P. (2009). Automatic meter classification in persian poetries using support vector machines. In *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*, pages 563–567. IEEE.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hopkins, J. and Kiela, D. (2017). Automatically generating rhythmic verse with neural networks. volume 1 of *ACL*, pages 168–178.
- Hu, J., Wang, G., Lochovsky, F., Sun, J.-t., and Chen, Z. (2009a). Understanding user’s query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web, WWW ’09*, pages 471–480, New York, NY, USA. ACM.
- Hu, X., Downie, J. S., and Ehmann, A. F. (2009b). Lyric text mining in music mood classification. *American music*, 183(5,049):2–209.
- Jamal, N., Mohd, M., and Noah, S. A. (2012). Poetry classification using support vector machines. *Journal of Computer Science*, 8(9):1441.
- Jang, M., Choi, J. D., and Allan, J. (2017). Improving document clustering by removing unnatural language. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, ACL.
- Jansen, B. J., Spink, A., Bateman, J., and Saracevic, T. (1998). Real life information retrieval: A study of user queries on the web. In *ACM SIGIR Forum*, volume 32, pages 5–17. ACM.
- Kaur, J. and Saini, J. R. (2017). Punjabi poetry classification: The test of 10 machine learning algorithms. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 1–5. ACM.

- Kazai, G. and Doucet, A. (2008). Overview of the inex 2007 book search track: Booksearch'07. In *ACM SIGIR Forum*, volume 42, pages 2–15. ACM.
- Kilner, K. and Fitch, K. (2017). Searching for my lady’s bonnet: discovering poetry in the national library of australia’s newspapers database. *Digital Scholarship in the Humanities*.
- Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., Speck, J. A., and Turnbull, D. (2010). Music emotion recognition: A state of the art review. In *Proc. ISMIR*, pages 255–266. Citeseer.
- King, G. and Zeng, L. (2001). Logistic regression in rare events data. *Political analysis*, 9(2):137–163.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the third International Conference on Learning Representations (ICLR'14)*.
- Korst, J. and Geleijnse, G. (2006). Efficient lyrics retrieval and alignment.
- Krallinger, M., Rabal, O., Lourenço, A., Oyarzabal, J., and Valencia, A. (2017). Information retrieval and text mining technologies for chemistry. *Chemical reviews*, 117(12):7673–7761.
- Kumari, K. P., Reddy, A. V., and Fatima, S. S. (2014). Web page genre classification: Impact of n-gram lengths. *International Journal of Computer Applications*, 88(13).
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.
- Lamport, L. (1994). *LATEX: a document preparation system: user’s guide and reference manual*. Addison-wesley.
- Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM.
- Leaman, R. and Gonzalez, G. (2008). Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*, pages 652–663. World Scientific.
- Li, H., Xu, G., Croft, B., and Bendersky, M. (2011). Query representation and understanding. In *Proceedings of the 2nd Workshop on query Representation and Understanding*. Citeseer.
- Li, X., Wang, Y.-Y., and Acero, A. (2008). Learning query intent from regularized click graphs. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08*, pages 339–346, New York, NY, USA. ACM.
- Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., Macdonald, C., and Vigna, S. (2016). Toward reproducible baselines: The open-source ir reproducibility challenge. In Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G. M., Hauff, C., and Silvello, G., editors, *Advances in Information Retrieval*, pages 408–420, Cham. Springer International Publishing.

- Lin, K. H.-Y., Yang, C., and Chen, H.-H. (2007). What emotions do news articles trigger in their readers? In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 733–734. ACM.
- Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.
- Liu, T.-Y. et al. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.
- Lopez, C., Partalas, I., Balikas, G., Derbas, N., Martin, A., Reutenauer, C., Segond, F., and Amini, M.-R. (2017). Cap 2017 challenge: Twitter named entity recognition. *arXiv preprint arXiv:1707.07568*.
- Lorang, E. M., Soh, L.-K., Datla, M. V., and Kulwicki, S. (2015). Developing an image-based classifier for detecting poetic content in historic newspaper collections. Technical report.
- Lowell, J. R. (1914). *Selected literary essays from James Russell Lowell*.
- McNamee, P. and Dang, H. T. (2009). Overview of the TAC 2009 knowledge base population track. In *TAC'09*, volume 17, pages 111–113.
- Metzler, D. and Croft, W. B. (2005). A markov random field model for term dependencies. In *SIGIR 2005*, pages 472–479. ACM.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Mitra, B. and Craswell, N. (2017). An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval (to appear)*. *Google Scholar*.
- Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. 29(3):436–465.
- Owoputi, O., O’Connor, B., Dyer, C., Gimpel, K., and Schneider, N. (2012). Part-of-speech tagging for twitter: Word clusters and other advances. *School of Computer Science*.
- Pass, G., Chowdhury, A., and Torgeson, C. (2006). A picture of search. In *InfoScale*, volume 152, page 1.
- Pennacchiotti, M. and Zanzotto, F. M. (2008). Natural language processing across time: An empirical investigation on italian. In *International Conference on Natural Language Processing*, pages 371–382. Springer.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Petrenz, P. (2014). Cross-lingual genre classification.

- Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR 1998*, pages 275–281. ACM.
- Promrit, N. and Waijanya, S. (2017). Convolutional neural networks for thai poem classification. In *International Symposium on Neural Networks*, pages 449–456. Springer.
- Rakshit, G., Ghosh, A., Bhattacharyya, P., and Haffari, G. (2015). Automated analysis of bangla poetry for classification and poet identification.
- Rather, C., Rather, L., Padgette, P., and Telser, R. (1994). *The Rather Press of Oakland, California*. Regional Oral History Office; The Bancroft Library; University of California.
- Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.
- Rayson, P., Archer, D., Baron, A., Culpeper, J., and Smith, N. (2007). Tagging the bard: Evaluating the accuracy of a modern pos tagger on early modern english corpora. In *Proceedings of the Corpus Linguistics conference: CL2007*.
- Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics.
- Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49. ACM.
- Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. Springer-Verlag New York, Inc.
- Rocchio, J. J. (1971). Relevance feedback in information retrieval.
- Rockwell, F., Loveless, A., and Hottes, A. (1917). *Garden Guide: The Amateur Gardener's Handbook*.
- Rose, D. E. and Levinson, D. (2004). Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM.
- Rosso, M. A. (2008). User-based identification of web genres. *Journal of the Association for Information Science and Technology*, 59(7):1053–1072.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.



- Schedl, M., Gómez, E., Urbano, J., et al. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends® in Information Retrieval*, 8(2-3):127–261.
- Scheible, S., Whitt, R. J., Durrell, M., and Bennett, P. (2011). Evaluating an ‘off-the-shelf’ pos-tagger on early modern german text. In *Proceedings of the 5th ACL-HLT workshop on language technology for cultural heritage, social sciences, and humanities*, pages 19–23. Association for Computational Linguistics.
- Sharoff, S. (2010). In the garden and in the jungle. In *Genres on the Web*, pages 149–166. Springer.
- Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. In *ACM SIGIR Forum*, volume 33, pages 6–12. ACM.
- Singhi, A. and Brown, D. G. (2014). Are poetry and lyrics all that different? In *ISMIR*, pages 471–476.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.
- Strauss, B., Toma, B., Ritter, A., de Marneffe, M.-C., and Xu, W. (2016). Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144.
- Swift, J. and Scott, S. W. (1824). *The Works of Jonathan Swift, D.D.* Archibald Constable and Co.
- Tizhoosh, H. R., Sahba, F., and Dara, R. (2008). Poetic features for poem recognition: A comparative study. *Journal of Pattern Recognition Research*, 3(1):24–39.
- Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*. ACL.
- Underwood, T. (2014). *Understanding Genre in a Collection of a Million Volumes*. University of Illinois, Urbana-Champaign.
- Underwood, T., Black, M. L., Auvil, L., and Capitanu, B. (2013). Mapping mutable genres in structurally complex volumes. In *IEEE Big Data*, pages 95–103.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.
- Veale, T. (2013). Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *ICCC*, pages 152–159.

- Vural, A. G., Cambazoglu, B. B., and Karagoz, P. (2014). Sentiment-focused web crawling. *ACM Transactions on the Web (TWEB)*, 8(4):22.
- Wakamiya, S., Kawai, Y., Kumamoto, T., Zhang, J., and Shiraishi, Y. (2015). Searching comprehensive web pages of multiple sentiments for a topic. In *Transactions on Engineering Technologies*, pages 337–352. Springer.
- Wang, D., Peng, N., and Duh, K. (2017). A multi-task learning approach to adapting bilingual word embeddings for cross-lingual named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 383–388.
- Won, M., Murrieta-Flores, P., and Martins, B. (2018). Ensemble named entity recognition (NER): Evaluating NER Tools in the identification of Place names in historical corpora. *Frontiers in Digital Humanities*, 5:2.
- Xiong, C. and Callan, J. (2015). EsdRank: Connecting Query and Documents through External Semi-Structured Data. In *CIKM'15*.
- Yalniz, I., Can, E., and Manmatha, R. (2011). Partial duplicate detection for large book collections. In *Proceedings of the 20th Conference on Information Knowledge and Management*, pages 469–474.
- Yan, R., Jiang, H., Lapata, M., Lin, S.-D., Lv, X., and Li, X. (2013). i, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *IJCAI*, pages 2197–2203.
- Yang, C., Lin, K. H.-Y., and Chen, H.-H. (2007). Emotion classification using web blog corpora. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pages 275–278. IEEE.
- Yang, X., Lin, X., Suo, S., and Li, M. (2017). Generating thematic chinese poetry with conditional variational autoencoder. *arXiv preprint arXiv:1711.07632*.
- Yang, Y. and Eisenstein, J. (2016). Part-of-speech tagging for historical english. In *Proceedings of NAACL-HLT*, pages 1318–1328.
- Yang, Y., Zhang, M., Chen, W., Zhang, W., Wang, H., and Zhang, M. (2018). Adversarial learning for chinese ner from crowd annotations. *arXiv preprint arXiv:1801.05147*.
- Yang, Y.-H. and Chen, H. H. (2012). Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):40.
- Yin, X. and Shah, S. (2010). Building taxonomy of web search intents for name entity queries. In *Proceedings of the 19th international conference on World wide web*, pages 1001–1010. ACM.
- Zamani, H. and Croft, W. B. (2016). Estimating embedding vectors for queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 123–132. ACM.

- Zamani, H. and Croft, W. B. (2017). Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 505–514. ACM.
- Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM.
- Zhang, X. and Lapata, M. (2014). Chinese poetry generation with recurrent neural networks. EMNLP, pages 670–680.
- Zhang, Y. (2015). Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 435–440. ACM.
- Zhang, Y. C., Séaghdha, D. O., Quercia, D., and Jambor, T. (2012). Auralist: Introducing serendipity into music recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12*, pages 13–22, New York, NY, USA. ACM.