Making Travel Smarter: Extracting Travel Information From E-Mail Itineraries Using Named Entity Recognition

Divyansh Kaushik*, Shashank Gupta+, Chakradhar Raju+, Reuben Dias+, and Sanjib Ghosh+

*Language Technologies Institute, Carnegie Mellon University
dkaushik@cmu.edu
+Amadeus IT Group
{shashank.gupta, mchakradharraju, reuben.dias, sanjib.ghosh}@amadeus.com

Abstract

The purpose of this research is to address the problem of extracting information from travel itineraries and discuss the challenges faced in the process. Business-to-customer emails like booking confirmations and e-tickets are usually machine generated by filling slots in pre-defined templates which improve the presentation of such emails but also make the emails more complex in structure. Extracting the relevant information from these emails would let users track their journeys and important updates on applications installed on their devices to give them a consolidated over view of their itineraries and also save valuable time. We investigate the use of an HMM-based named entity recognizer on such emails which we will use to label and extract relevant entities. NER in such emails is challenging as these itineraries offer less useful contextual information. We also propose a rich set of features which are integrated into the model and are specific to our domain. The result from our model is a list of lists containing the relevant information extracted from ones itinerary.

1 Introduction

We are progressing towards a fully digital and paperless world wherein electronic documents and emails are increasingly becoming more secure and trusted mode of communication. Email has become one of the most popular modes of communication also because of the high adoption of smart and mobile devices that allow email access virtually from anywhere. As a result, an increasing number of businesses are adopting emails as their preferred Business to Consumer (B2C) communication channel to share personalized and sensitive information with their customers. These emails are usually machine generated and composed by filling information from databases into slots in predefined templates and often include personalized greetings and other information. Though these kind of templates enhance presentation of the email, the user is primarily interested in the data relevant to the business.

One such document is the Travel Itinerary sent across by a travel agency or an airline confirming the travel. Travel itineraries vary a lot from simple itineraries in plain text format to tabular itineraries in raw text as well as rich text formats and might contain more than one table or nested tables as well. Extraction of the relevant content from these itineraries and presenting it in a simple structural format would be of great convenience, especially for business travelers who travel very often and have a busy daily routine. In the past, extraction of information has been tried on similar kind of data by the use of wrappers or parsers (Hall et al., 2011; Crescenzi et al., 2001). Since these emails potentially contain personal information, passenger record locater(s) etc. it is essential to maintain the privacy of the traveler while trying to accurately extract the correct information. This problem has been tackled by organizations in the industry by creating parsers for each kind of template used by the travel providers. With more than 1000 registered commercial airlines, varying formats and frequent personalization done by airlines or third party aggregators like Skyscanner or Expedia, it becomes hard to maintain this ever growing number of templates as it demands a significant amount of resources, time and manual effort. This kind of approach is not scalable and a slight change in one template would cause the parser to fail.

First two authors had equal contribution. Work was done when all authors were at Amadeus IT Group.

Proceedings of Recent Advances in Natural Language Processing, pages 354–362,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_047

Itinerary

Carrier	Flight #	Departing	Arriving
American Tom Cruise	2379 Seat 22E	MIAMI INTERNTNL SAT 22OCT 5:55 PM Economy	ST THOMAS 8:39 PM FF#: AA123456
American Tom Cruise	1350 Seat 26E	ST THOMAS SAT 29OCT 8:30 AM Economy	MIAMI INTERNTNL 11:27 AM FF#: AA123456

Figure 1: Snapshot of a plain text travel itinerary that looks like a tabular itinerary.

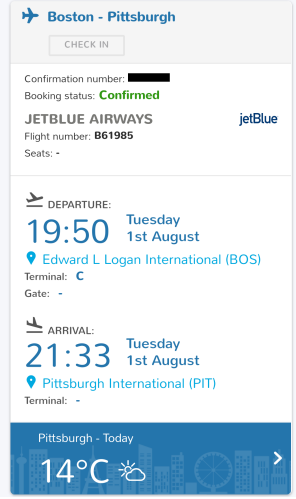

Figure 2: Structured information presentation in applications making use of regular expressions. Flight number, carrier information, boarding and off point, terminal information along with date and time has been extracted. These applications provide real time update to user′s journey.

In this paper, we discuss an approach based on Named Entity Recognition to "parse" such itineraries. For the purpose of this paper, we have considered our scope to be limited to emails that contain flight segments and are in English language. Since we have defined our domain very specifically and we know what we need to extract while also knowing to some extent what kind of data we should expect in an email, a supervised learning approach seems fit. Use of semi-supervised or unsupervised approaches is not feasible at the time given the small amount of data that we have. We investigate the application of Named Entity Recognition (NER) technique towards extracting information from B2C emails in the domain of air travel. The challenge in front of us was not only to extend the application of these NLP techniques to an entirely new domain but also to gather the data and work on it while still following all the laws applied to maintaining one′s privacy. Here, NER is used to label and extract entities like airport, person′s name, location, flight, carrier etc. This will be followed by validating the extracted entities with the open travel data (Arnaud, 2017). The contribution of this paper is twofold: investigate application of named entity recognition on the domain of air travel itineraries and use it to extract information, and present domain specific features to improve the performance of the NER. The end result from our model contains information about various journey legs as extracted from the email after the process of tagging and validation. This concept of NER on restricted domains has been previously used in domains like molecular biology, bioinformatics, and security etc. but extending it to B2C emails is a challenging task. B2C emails especially travel itineraries offer little contextual evidence for the interpretation of the terms that appear in the text. State of the art approaches for named entity extraction from well-edited text rely heavily on orthographic features such as capitalization and part of speech (POS) tagging which are unreliable in case of travel itineraries thereby waning the usefulness of such features for NER in this domain and posing further challenges towards robust entity extraction.

We make use of a Hidden Markov Models which are already verified to be mature and trustworthy on refined text (Zhou and Su, 2002; Minkov et al., 2006b; Ekbal and Bandyopadhyay, 2007; Zhao, 2004) to develop our NER. Our HMM model which when combined with our custom features gives us an F1 value of 70.2% upon experimentation.

2 Related Work

Information extraction from emails is a very complex task. It shares, however, a lot in common with information extraction from the web. Both contain some amount of HTML content which is the very basic similarity. Much work has been done to exploit the HTML structure to extract information from web pages. Buttler et al.(2001) presented rules based on the tree representation of web documents and extracting content from web pages. Crescenzi et al.(2001) presented an approach to extract data from HTML sites based on comparison of HTML pages and then generating wrappers based on their similarities or differences. Their model did not have any predefined knowledge of the organization of web pages which was not the case with either WIEN or STALKER which gener-

ated their wrappers by examining a number of labeled examples. Banko et al.(2007) presented the Open IE paradigm which did not require any human input and worked towards extracting relation tuples over a single data driven pass over the corpora, however, it does not build any coherent sets of category or relation instances as constructed by Dalvi et al.(2012) in which concept-instance pairs were extracted from HTML corpus. Their method relied on clustering terms found in HTML tables which was followed by assignment of concept names based on Hearst patterns. A Named Entity Recognition system was also presented for complex named entities in web text which made use of a lexical statistics based approach (Downey et al., 2007). Paşca et al.(2006) applied the NER technique to very large web corpora and demonstrated promising results from a very small seed of example facts. Similarly, Zhang et al.(2015) made use of a semi-supervised approach by making a combination of a weak learner(EM) with plain CRFs. Due to the privacy concerns, work on emails has been limited due to public availability of a few corpora. Klimt and Yang(2004) presented the Enron email corpus for email classification research. Minkov et al.(2005) worked on this dataset towards extraction of personal names from emails. They compared results from a Voted Perceptron HMM and a CRF model. Minkov et al.(2006b) worked towards adjusting the recall-precision tradeoff in NER systems as per user performance criteria. They presented their work on the Enron dataset (Klimt and Yang, 2004) along with other datasets. Major work on emails has been on binary classification of emails as spam or not spam (Koprinska et al., 2007; Youn and McLeod, 2007), and also towards using classification (Klimt and Yang, 2004; Bekkerman et al., 2004; Dredze et al., 2006) and clustering methods (Huang et al., 2004; Huang and Mitchell, 2006; Li et al., 2006) for categorizing emails into folders. Carvalho and Cohen(2004) worked towards learning to extract signatures and reply lines from plain text emails while Minkov et al.(2006a) also worked towards disambiguating names in emails by making use of graph-walk similarity measures. Besides the works on content mining in emails, there have been some works towards event extraction (Nelson et al., 2014) and prioritizing emails for improved user experiences (Eugene and Caswell, 2015; Laclavik and Maynard, 2009). Our data varies a lot from all these datasets because of the numerous formats in which travel itineraries can come. Also, we cannot make use of features like POS tags or capitalization information as these emails have minimal useful contextual information and are many-a-times present in all caps. This makes our problem statement different from the works that have been done before.

3 Data, Privacy and Characteristics

It is important for businesses to maintain user privacy while sending emails, not just because of the privacy laws - which are very strict and hold businesses to very high principles - but also because of the policies of privacy and ethics such organizations set. Such policies are to ensure that users personal information will not fall into the hands of any third party. This makes it difficult to generate an annotated training set which fares well on the fronts of diversity and complexity as one would encounter in the real world. To counter this, we created a dataset of 2,000 training emails and 600 test emails, none of which correspond to any individual. These were synthetically designed by Amadeus IT Group for testing the regression suites on their email parsing products. These emails have been chosen at random to ensure diversity and complexity. The only difference between emails in our dataset and real world emails is that the data in our emails does not correspond to any real individual. Doing so, we were able to protect one′ privacy. One might wonder what differences are found between our data and well edited data like the newswire corpus on which the performance of NER techniques have been well established. For this purpose we have outlined key characteristics of named entities in our domain along with the emails they are present in to get an understanding of the domain before describing our solution:

- Most emails occur in all caps and have very less usable contextual information making it hard to extract POS features.
- Named entities may have different representations. *John F. Kennedy International Airport, JFK International and JFK Intl. Airp.* refer to the same entity. Similarly *British Airways Flight 9* is same as *BA9* or *BA 9*.
- Sometimes due to character space restraints, an entity may not appear in its complete form.

Entity	Description	Examples
Name	Passenger Name	Tom Cruise, CRUISE/TOM
PNR (Passenger Name Record)	6 character record locater	CC1ORJ, ABC123
Booking Reference	Non 6 char. record locater	S23VG, C7GH1YV
Date	Departure/Arrival date	21-08-2017, 21AUG17
Time	Departure/Arrival time	06:55PM, 1855 hours, 18:55
Airport	Airport Name	Logan International Airport
Location	City of the said airport	Boston
Airport Code	IATA code of the airport	BOS
Carrier	Name of airline	Alitalia, American Airlines
Flight	Carrier name/code followed by flight number	American Airlines 5526, LH400, BA 9

Table 1: Entity classes for the travel itineraries targeted for the probabilistic HMM based NER.

We may see *O Hare Interna* in place of *O Hare International* due to these restraints.

- Entities may appear in a cascaded form like *McCarran Las Vegas International Airport* where *McCarran International Airport* is the airport of *Las Vegas* which is a city. More efforts and validation processes need to be made to identify and extract such entities.

- Two entities might be similar in resemblance. *Indira Gandhi* may be a person traveling from *Indira Gandhi International Airport*. Also, *VS2491* is a *Virgin Atlantic* flight from *New York* to *Pittsburgh* but it might be a PNR as well. It is hard to resolves such ambiguities.

- One named entity may share two or more head nouns for eg: *Envoy Air as American Eagle, Flight 3530 from Buffalo to Chicago*. Here *Envoy Air* and *American Eagle* are two head nouns for flight number *3530*.

Further, the emails may appear in a tabular format without any headers in which case there is usually minimal contextual information for the NER to learn. Such cases make NER difficult on such kind of data. Due to this reason we explore various features that could help us identify such named entities in the text.

4 Methods and Features

4.1 Hidden Markov Model

Emails are processed such that each word in an email is passed as a separate token to an HMM to find and extract the tokens which are part of the entities of interest. In the task of Named Entity Recognition, the hidden state can be thought of as the sequence of tokens while the output sequence is the statistically optimal sequence of labels corresponding to the input word sequence.

Our HMM is inspired from Zhou and Su(2002). The mathematics of their model could be described as follows: Let a sequence of tokens be:

$$S_1^n = S_1 S_2 S_3 S_4 .... S_{n-1} S_n \tag{1}$$

the objective of the NER is to find the statistically optimum tag sequence

$$T_1^n = T_1 T_2 T_3 T_4 .... T_{n-1} T_n \tag{2}$$

by trying to maximize the probability P($T_1^n|S_1^n$). Each s_i is defined as $\langle f_i, w_i \rangle$ where w_i is the ith word in the sequence and f_i is the set of features attributed to w_i. We follow the BIO notation for labeling and each of our t_i is structurally composed of two parts: the entity class and the feature set. The entity class can be one of the defined classes while labeling the training set along with the OTHER class. The feature set is added in order to represent more accurate and precise models based on the limited number of label boundary categories and entity classes.

From their model, P($T_1^n|S_1^n$) can be represented as:

$$logP(T_1^n|S_1^n) = logP(T_1^n) + log\frac{P(T_1^n, S_1^n)}{P(T_1^n)P(S_1^n)} \tag{3}$$

Assuming mutual independence in the second term, we have:

$$log\frac{P(T_1^n, S_1^n)}{P(T_1^n)P(S_1^n)} = \sum_1^n log\frac{P(t_i, S_1^n)}{P(t_i)P(S_i^n)} \tag{4}$$

Hence we have,

$$logP(T_1^n|S_1^n) = logP(T_1^n) - \sum_1^n logP(t_i) + \sum_1^n logP(t_i|S_1^n) \quad (5)$$

by substituting equation 4 in equation 3. Now, we can calculate the first term on the RHS of equation 5 by using the chain rule of probability which allows us to compute joint probabilities by making use of only the conditional probabilities. Each term in our case is assumed to be dependent on the previous two terms (trigram modeling). The second term is the log probability of all label occurrences. We apply the Viterbi algorithm (Forney, 1973; Viterbi, 1967) which makes use of a dynamic programming approach to find the most likely tag sequence for our given sequence of tokens based on the state transition probabilities of the tags and emission probability of s_i given t_i. The emission probability has been smoothened using Lidstone′s Law which can be thought of as

$$e_k(s_j) = \frac{N_k(s_j) + \lambda}{\sum_p N_k(s_p) + |NV|}$$

where $N_k(s_j)$ is the number of times a token s_j is emitted in the state k in the training set. λ is a constant whose value lies between 0 and 1 and can be used to vary the degree of discounting offered in smoothing and $|NV|$ is the number of distinct word types in the training set. Similarly smoothing with λ is also applied to the transition probabilities.

4.2 Features

The features defined for this model are domain specific and effort has been put in to craft features that are useful in identifying particular classes. This not only boosts the results of our baseline model but also provides an insight into how specific features effect a particular class. Our features can be classified into three major categories:

1. *Orthographic Features:* In this kind, we have devised special features which are designed to capture the word formation. For eg:- a 6 digit alpha-numeric value which appears in an email in all caps is most likely a passenger record locator. Similarly, an all alphabet token with / in between is most likely to be a name in the format of *<last name>/<first name>/MR* as it appears on your boarding pass. Similarly, any token which is an a single alphabet followed by a dot(.) is most likely to be part of an entity name - person′s name, airport name, country name etc. A list of such features has been given in Table 2.

2. *Contextual Features:* Features which are based on the previous word or words can be designed on the fly by our model while learning the training set. These features are identified using specially crafted bigrams and tri-

Feature Name	Example
Common word (StopWord)	from, in
Abbreviation with dot	J.F.K., Jr.
All alphabets	John
Alphabets and Slash - (slash not at extreme)	Paul/John
One Digit	9
Number with length ≤ 6	12
Digit dot digit	1.259
2 char. alphanumeric - slash 6 char. alphanumeric	LX/UV231Y
Digit slash digit	23/10/2016
Digit hyphen digit	23-10-2016
6 character alphanumeric	AB23C5

Table 2: Orthographic Features.

Dictionary Features	Example
Word is in airline name	Lufthansa
Word is in airport name	Port Bouet Airport
Abbreviation is - airline IATA code	LH
Abbreviation is - airline ICAO code	DLH
Abbreviation is - airport code	ABJ
Word is in airport loc.	Abidjan
Word is in alternate - airport name dictionary	Houphouet-Boigny International Airport
Word is in alternate - airline name dictionary	Deutsche Lufthansa
Flight no. in routes dict.	LH404

Table 3: Dictionary Features. All these features are used in the vector creation process as well as the validation process.

grams which are of the form <*pr. word, current word tag*> and <*pr. pr. word, pr. word, current word tag*>. These are used to identify most likely patterns that could be found for instances such as names, PNRs etc. We identify such patterns and assign a confidence value to such patterns based on the formula:

$$C = \frac{N(correct) - N(incorrect)}{N} \quad (6)$$

where N(Correct) is the total occurences of the pattern when it forms the context of a word having the label of interest and N(incorrect) is the total occurences of such patterns when followed by the label which is not of interest. We generate such patterns on the go for identifying entities such as *Name(s), PNR(s)* and *Booking Reference(s)*. This equation is based on the assumption that such word patterns are useful. After generating all such patterns we make use of a threshold of 60% to shortlist the features of use. Patterns common for a particular entity type are clubbed together as a feature to avoid the problem of data sparseness.

3. *Dictionary features:* We make use of several dictionaries as features. These include airline dictionary, airport dictionary and a dictionary of routes, which not only reflect on the airport/airline names but also allow us access to other information, helping us validate our tags and improve precision. Our dictionary of airports and airlines was obtained from the open travel data repository (Arnaud, 2017). The airport dictionary constructed from this data consists of airport names along with their codes, alternative names, location etc. Similarly our airline dictionary includes airline name, alternative names, IATA and ICAO codes. These help us when there are multiple names for the same airport or the airline and also in deciding whether an abbreviation is an airline code, airport code or none. The routes dictionary consists of information of all the flights operating in the year 2016-17. Not just are these used in feature vector creation but are also used in our later validation stage where we match our airports, flight numbers etc. from the routes data to validate the tags given by the HMM tagger. A few features are listed in Table 3. In validation stage, while matching *Airport* and *Carrier* entities with dictionary entries we make use of regular expression search (to match partial names as well) using tail recursion for good performance.

5 Experimental Results

Our system was trained on a training set of 2,000 randomly chosen emails which were manually tagged using the BIO tag notation. The tests were performed on a set of 600 randomly chosen e-mails. We evaluate our NER tagger on the metrics of Precision, Recall and F1 Measure. We report the entity-wise results as well as the overall results. Note that the entity-wise results are important for our validation stage which use the same dictionaries mentioned earlier. Table 4 shows our results on various feature sets (the base model remains the same in all cases). The highest score in each category of Precision, Recall and F1 Measure has been written in bold face. entity-wise results are important because of various reasons:

- For entities such as *Name*, *PNR* and *Booking Reference* which do not have any dictionaries to be validated against, we need high precision. A high recall would be ideal but many times these entities occur multiple times in the same email but have the same value. It would be ideal if the NER tags all the occurrences of a *PNR* as *PNR* but tagging one occurrence of that particular value as PNR is sufficient for our extraction process.

- For entities such as *Airport*, *Location* and *Carrier*, a validation process is present where use of dictionaries is made. Because of this, a high recall is required. We rule out the incorrectly tagged airports, locations etc. by matching them in the validation dictionaries.

- For entities like *Flight Number* and *Airport Code*, we can validate the journey legs on their basis. We require a high precision as well as recall on these entities to verify and validate the journey legs of an individual.

It can be seen that our baseline features and combination of those features have been able to improve the results of our system. As a general trend, entities involving names (*Name*, *Airport*) have a low F1 measure of being labeled. One reason for this is that these entities are usually longer than any

Model	Overall	Name	PNR	BRef	Date	Time	ACode	Airp	Loc	Carr	Flt
BASE	0.60	0.31	0.59	0.70	0.72	0.75	0.67	0.45	0.54	0.65	0.31
ORTH	0.66	0.39	0.70	0.83	**0.83**	0.76	0.78	0.46	0.58	0.65	0.55
CTXT	0.63	0.59	0.87	0.73	0.72	0.70	0.73	0.44	0.56	0.65	0.21
DICT	0.62	0.32	0.60	0.70	0.74	0.77	0.93	0.32	0.28	0.80	0.80
O+C	0.69	**0.63**	0.88	0.84	0.80	0.76	0.77	**0.59**	**0.62**	0.65	0.31
O+C+D	**0.70**	0.62	**0.91**	**0.89**	0.82	**0.78**	**0.95**	0.41	0.38	**0.87**	**0.95**

Table 4: F-measure results on the test dataset. The highest values in each column are written in bold. ORTH(O):Orthographic, CTXT(C): Contextual, DICT(D): Dictionary features.

Model	Precision	Recall	F Measure
Base	0.67	0.54	0.60
O+C	**0.72**	0.66	0.69
O+C+D	0.70	**0.70**	**0.70**

Table 5: Overall results on various models.

Entity	Precision	Recall
Name	0.65	0.60
PNR	0.91	0.90
Booking Reference	**0.96**	0.83
Date	0.83	0.81
Time	0.81	0.76
Airport Code	**0.96**	**0.95**
Airport (Name)	0.34	0.53
Location	0.66	0.27
Carrier	0.87	0.86
Flight	**0.96**	0.94

Table 6: Entity-wise Precision and Recall values on final system (O+C+D).

other entity type and not all tokens get the correct labels. The F measure of *Airport Codes, Booking References, PNR* and *Flight* has improved exceptionally with the introduction of features. Also, we can see that *Booking Reference, Flight* and *Airport Code* have the highest precision while *Airport Code, Flight, PNR* and *Carrier* have the highest recall. We haven′t used a *Name* dictionary yet because even though our emails are in English, the passenger might be from a region for which such data is not available. We have also analyzed that since the airport names are longer than other entity types, they are have a lesser likelihood to match the dictionary entries for instance: *MIAMI INTERNTNL* as shown in Figure 1 will not match with the dictionary entry of *Miami International Airport*. Similarly *O Hare Intl Airp* will not match with the entire dictionary entry of *O′Hare International Airport* and the maximum subsequence match will only match *Hare*. This diminishes the value of the *Airport* dictionary and we will discuss this limitation as part of future work. For airlines, since the names are smaller, they match very well with the *Carrier* dictionary entries. Correctly extracting the *Airport Codes* helps us mitigate the poor results shown by the *Airport* entity type. Figures[1] also show extracted journey legs from various itineraries.

6 Conclusion and Future Work

In this paper, we have shown how a probabilistic NER can be applied to B2C emails with focus on travel itineraries and extract relevant data from the same. The emails were chosen randomly to ensure the unpredictability of the format of data representation. We trained our system on the small number of emails that were donated to us for this research and proposed various domain specific features while not using features like POS tagging, capitalization etc. The resulting system was able to extract desired information from test emails with high accuracy and efficiency.

Upon analyzing our results, we have found that the dictionaries are too rigid and we need to find a way to make sure that various representations of the same entity and its aliases are taken care of while matching test data with dictionary entities, especially in the case of *Airport* dictionary. We also plan to improve *Name* extraction by making use of various features as presented by Minkov et al.(2005). With this, we also believe that there is a need to explore CRF and RNN based NER models and compare how each model works. Since our dataset is relatively small, it is also of wondering what impact will increasing the training set have on our system′s performance. We intend to gather more training data and evaluate our system′s performance as the data increases

[1] https://goo.gl/LtvuCb

References

Denis Arnaud. 2017. Open travel data. Amadeus IT Group. https://github.com/opentraveldata/opentraveldata.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'07, pages 2670–2676. http://dl.acm.org/citation.cfm?id=1625275.1625705.

R. Bekkerman, A. McCallum, and G. Huang. 2004. Automatic categorization of email into folders: Benchmark experiments on enron and sri corpora. *Center for Intelligent Information Retrieval, Technical Report IR* 418.

D. Buttler, Ling Liu, and C. Pu. 2001. A fully automated object extraction system for the world wide web. In *Proceedings 21st International Conference on Distributed Computing Systems*. pages 361–370. https://doi.org/10.1109/ICDSC.2001.918966.

Vitor R. Carvalho and William W. Cohen. 2004. Learning to extract signature and reply lines from email. In *In Proceedings of the Conference on Email and Anti-Spam*.

Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. 2001. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the 27th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, VLDB '01, pages 109–118. http://dl.acm.org/citation.cfm?id=645927.672370.

Bhavana Bharat Dalvi, William W. Cohen, and Jamie Callan. 2012. Websets: Extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, USA, WSDM '12, pages 243–252. https://doi.org/10.1145/2124295.2124327.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'07, pages 2733–2739. http://dl.acm.org/citation.cfm?id=1625275.1625715.

Mark Dredze, Tessa Lau, and Nicholas Kushmerick. 2006. Automatically classifying emails into activities. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*. ACM, New York, NY, USA, IUI '06, pages 70–77. https://doi.org/10.1145/1111449.1111471.

Asif Ekbal and Sivaji Bandyopadhyay. 2007. A hidden markov model based named entity recognition system: Bengali and hindi as case studies. In *Proceedings of the 2Nd International Conference on Pattern Recognition and Machine Intelligence*. Springer-Verlag, Berlin, Heidelberg, PReMI'07, pages 545–552. http://dl.acm.org/citation.cfm?id=1781034.1781108.

Louis Eugene and Isaac Caswell. 2015. Making a manageable email experience with deep learning.

G. D. Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE* 61(3):268–278. https://doi.org/10.1109/PROC.1973.9030.

Keith B. Hall, Ryan T. McDonald, Jason Katz-Brown, and Michael Ringgaard. 2011. Training dependency parsers by jointly optimizing multiple objectives. In *EMNLP*. ACL, pages 1489–1499. http://www.aclweb.org/anthology/D11-1138.

Yifen Huang, Dinesh Govindaraju, Tom M Mitchell, Vitor Rocha de Carvalho, and William W Cohen. 2004. Inferring ongoing activities of workstation users by clustering email. In *CEAS*.

Yifen Huang and Tom M. Mitchell. 2006. Text clustering with extended user feedback. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '06, pages 413–420. https://doi.org/10.1145/1148170.1148242.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Proceedings of the 15th European Conference on Machine Learning*. pages 217–226.

Irena Koprinska, Josiah Poon, James Clark, and Jason Chan. 2007. Learning to classify e-mail. *Inf. Sci.* 177(10):2167–2187. https://doi.org/10.1016/j.ins.2006.12.005.

Michal Laclavik and Diana Maynard. 2009. Motivating intelligent e-mail in business: An investigation into current trends for e-mail processing and communication research. *2009 IEEE Conference on Commerce and Enterprise Computing* pages 476–482.

H. Li, D. Shen, B. Zhang, Z. Chen, and Q. Yang. 2006. Adding semantics to email clustering. In *Sixth International Conference on Data Mining (ICDM'06)*. pages 938–942. https://doi.org/10.1109/ICDM.2006.16.

Einat Minkov, William W. Cohen, and Andrew Y. Ng. 2006a. Contextual search and name disambiguation in email using graphs. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '06, pages 27–34. https://doi.org/10.1145/1148170.1148179.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '05, pages 443–450. https://doi.org/10.3115/1220575.1220631.

Einat Minkov, Richard C. Wang, Anthony Tomasic, and William W. Cohen. 2006b. Ner systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-Short '06, pages 93–96. http://dl.acm.org/citation.cfm?id=1614049.1614073.

M.F. Nelson, L. Tannenbaum, D. Bhowal, and M. Sharma. 2014. System and method for extracting calendar events from free-form email. US Patent 8,832,205. http://www.google.com/patents/US8832205.

Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL-44, pages 809–816. https://doi.org/10.3115/1220175.1220277.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13(2):260–269. https://doi.org/10.1109/TIT.1967.1054010.

Seongwook Youn and Dennis McLeod. 2007. *A Comparative Study for Email Classification*, Springer Netherlands, Dordrecht, pages 387–391.

Weinan Zhang, Amr Ahmed, Jie Yang, Vanja Josifovski, and Alex J. Smola. 2015. Annotating needles in the haystack without looking: Product information extraction from emails. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '15, pages 2257–2266. https://doi.org/10.1145/2783258.2788580.

Shaojun Zhao. 2004. Named entity recognition in biomedical texts using an hmm model. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*. Association for Computational Linguistics, Stroudsburg, PA, USA, JNLPBA '04, pages 84–87. http://dl.acm.org/citation.cfm?id=1567594.1567613.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 473–480. https://doi.org/10.3115/1073083.1073163.