Spacy: The Final Frontier

Mark Hutchens
Brandeis University / Waltham, MA
mhutchens@brandeis.edu

Samantha Richards
Brandeis University / Waltham, MA
srichards@brandeis.edu

Lucino Chiafullo
Brandeis University / Waltham, MA
lrchiafullo@brandeis.edu

Emily Fountain
Brandeis University / Waltham, MA
emilyfountain@brandeis.edu

Abstract

The *Star Trek* fandom constitutes a vast and diverse set of corpora, both canonical and fan-generated. We pursued the task of training a crfsuite ML model to extract named entities from a dataset consisting of transcripts of Stark Trek episodes, then applying that model to the domain of fan fiction in order to investigate the generalizability of the model.

1 Introduction

When approaching this project, we were particularly attracted to the idea of annotating data from a vast fictional universe and performing entity extraction on that data. *Star Trek*, the popular science-fiction franchise consisting of TV shows, movies, and books, fulfilled many of our needs. To begin with, it has unique fictional entities, and there is a large amount of relevant linguistic data available online (both in the form of primary documents such as scripts, and secondary sources like fan-written stories). Additionally, many of the same entities appear throughout the long timeline of the series (between multiple television series, throughout novels, etc.). Therefore, the goal of the project was to pull data from multiple modalities such as scripts, fan fiction, books, forums, and articles, train a model similar to the one we had previously implemented on the data, and test the model on the different modalities of data. Time constraints led to a paring down of the data sources, so that we were left with television scripts and online fan fiction.

2 Relevant Work

When devising our annotation scheme, the ACE Annotation Guidelines for Entities were particularly helpful. We based many of our types on the existing framework from ACE, such as PER, LOC, ORG, and GPE. SHIP was also inspired by the Facility tag in the ACE guildelines. As is detailed in the following sections, even where we took entity types wholesale from the ACE guidelines, we changed the definitions to fit our needs and datasets.

3 Data and Corpus

Our data is derived from two main sources, http://chakoteya.net/ and https://trekfanfiction.net/. The former is the source for transcripts, the latter for fanfiction.

http://chakoteya.net/
https://trekfanfiction.net/

To pull transcripts from chakoteya, we used the Python package BeautifulSoup. The structure of the website was fairly predictable, so with some careful string matching we were able to scrape about 700 episodes from The Original Series through Enterprise. That should be all of them.

We annotated the first 50 lines from each of the episodes, leaving in stage directions and other descriptive elements. Our reasoning was that the start of episodes would give many of the named entities for the rest of the episode, while allowing us to cover the entire breadth of the series.

To create a fan fiction dataset, 30 stories were selected from https://trekfanfiction.net/. Stories about each series were selected, so that all shows represented in the television show would be present in the fan fiction data. These stories were each stored in a text file, which were themselves split into sections of 30 lines. The resulting

	Ann1	Ann2	Ann3	Ann4
Ann1	1	0.4503	0.4136	0.4382
Ann2		1	0.5877	0.5764
Ann3			1	0.6603
Ann4				1

Table 1: Cohen's kappa among four annotators on trial 33 episodes

files were split between three batches that could be annotated using Doccano.

4 Annotation

Our complete annotation guidelines are in Appendices A and B, but here is a brief introduction to the tags we chose:

1. PER - Individual persons, by name or by rank
2. SHIP - Named ships, shuttlecraft
3. LOC - Planets, quadrants, starbases, locations within a ship
4. SPEC - All species, animal and humanoid. Includes monoculture empires (such as the Xindi)
5. ORG - Includes groups of people working on a specific part of the ship (like Engineering), and other groups like Vulcan High Council, Astronomical Survey Team, etc.
6. GPE - Groups with a populace of citizens, like The Federation of Planets.

We had considered using TECH for technologies, but eventually ruled it out as too ambiguous. We also considered WARP for the various warp speeds, and WPN for weapons, but ruled these out because they were considered too sparse.

We began with a preliminary annotation run of 33 episodes in order to calculate inter-annotator agreement (IAA). The results of this are shown in Table 1. We had an extended discussion about how to resolve these differences in a Facebook group chat, though the exact proceedings are difficult to export.

The remaining episodes were divided roughly equally between the four annotators, which each annotator receiving an equal number of episode segments from each series (in order to balance the entity mentions that may be encountered).

5 Entity Extractor

5.1 Entity extraction pipeline

Our entity extraction pipeline began with a corpus of json objects, housed in a single .jsonl file, each representing a single annotated document (that is, first 50 lines of a *Star Trek* episode). Using spacy's en_core_web_sm, we processed these into spacy docs, which included tokenizing the text into spacy tokens and storing the entity mentions as spacy span objects. A small amount of pre-preprocessing was necessary before the spacy docs could be created: this mainly consisted in removing newline characters and handling a few instances of adjacent bracket characters, such as ")[", which would have led to incorrect tokenization in a small number of cases. However, for the most part, the data was quite clean.

After preprocessing, we split the dataset into a train and a dev set. As mentioned above, we had 706 annotated documents in total, and we chose a train/dev split of roughly 80:20, resulting in 565 train docs and 141 dev docs. We did not include a test set in this split, as we intended to test our model on our fan fiction dataset.

For the model itself, we used a crfsuite tagger. To prepare the spacy docs for training, we implemented a set of feature extractors, which extracted from each train set doc a set of relevant features for the model to use over a specified context window. We also implemented our own entity encoders, which applied a specified encoding scheme (IO, BIO, or BILOU) to the spacy entity spans within each doc. These components of the pipeline allowed us the flexibility to easily modify the feature set and encoding scheme in order to investigate the optimal combination. Because the call to crfsuite's trainer permits selection from multiple learning algorithms, we were able to flexibly experiment with this as a setting as well. In the subsequent sections we review the results of that process.

After training the crf suite tagger using a set of features extracted from the train set docs, we applied the trained model to an unlabeled version of the dev set, generating a set of spacy docs containing predicted entity types. We evaluated this against a gold dev set containing the original labels and collected precision, recall, and F1 scores for each entity category as well as overall.

5.2 Experimental design

In order to determine the optimal encoding scheme, feature set, and learning algorithm, we conducted a series of experiments, using the resulting F1 against the dev set as the main performance metric. We began with an ablation study on a standard feature set. We also experimented with more 'advanced' features, namely Word2Vec embeddings. After the optimal feature set had been determined, we experimented with three different encoding schemes. Finally, we tested several different crfsuite learning algorithms. The details of the experiments and their results are discussed below.

5.3 Feature selection

We first tested the model with the following set of baseline features, which were extracted for the target token as well as every context token within a specified window size:

- Bias: only included if target token
- Token: the string of the token
- UpperCase: all characters are upper case
- TitleCase: first character is uppercase
- InitialTitleClase: first character is uppercase and token is sentence-initial
- Punctuation: token contains punctuation
- Digit: token contains a digit
- WordShape: word shape of the token

In addition to these features, this first pass included a window size of 1, BILOU as the encoding scheme, and averaged perceptron as the learning algorithm.

The results of the baseline feature set are included in Table 2. The model did quite well, yielding an F1 of 78.60 overall. It performed the best on PER, with an F1 of 84.81, and performed the worst on GPE, with an F1 of 60.67. This is not surprising, given that the PER entity type was the one most frequently tagged by our annotators (6177 annotated mentions, see Table7), while GPE was both the most infrequently tagged (only 193 annotated mentions) and the most inconsistently tagged (tagged only twice by annotator D, but between 54 and 66 times by the remaining annotators). See the discussion section below for further notes on this.

Type	Prec	Rec	F1
ALL	80.81	76.50	78.60
GPE	69.23	54.00	60.67
LOC	74.22	70.40	72.26
ORG	75.00	59.48	66.35
PER	85.32	84.31	84.81
SHIP	80.18	64.03	71.20
SPEC	77.23	72.08	74.57

Table 2: Baseline Results - Dev set

Next we selectively ablated each of the features to observe its impact on the performance of the model. Interestingly, the isolated ablation of bias, uppercase, punctuation, digit, and wordshape features each resulted in in slight improvements over the baseline, with wordshape feature bringing the largest improvement, from 78.40 to 79.09.

In addition to the ablation studies, we experimented with implementing additional features to capture subword information, namely the following three:

- Prefix: first two characters if token is longer than 4 characters
- Suffix: last three characters if token is longer than 4 characters
- Stem: the word stem when passed through the Porter stemmer

None of these features in any combination had any impact on the performance, which held firmly at an overall F1 of 79.09.

Finally, we experimented with several higher order features. Our first stop was Brown clusters, since our previous experience with the entity extraction from the Yelp restaurant review dataset indicated that they might provide some help. We found several implementations of Brown clusters on Github; however, they turned out to be either difficult to use, extremely slow, or simply incorrect. Given more time, our team would have liked to implement Brown clusters for this dataset ourselves, but not having sufficient time, we were forced to scrap the effort.

Fortunately, more reliable implementations exist for word embeddings than for Brown clusters, and so we were able to more successfully incorporate and test this feature. We trained a gensim Word2Vec model on a corpus consisting of the full text of all Star Trek episodes from Enterprise, The

Original Series, Voyager, The Next Generation, and Deep Space Nine (the reader is reminded that only the first 50 lines of these episodes were part of the annotated corpus). The inclusion of this feature, with a scaling of 1.0, resulted in a modest improvement, yielding an F1 of 81.38. Scalings of 0.5, 1.5, and 2.0 also led to slightly smaller improvements. Lastly, when the Word Vector feature (1.0 scaling) was tested with the WordShape feature turned off, the resulting performance decreased to 80.79. Though perplexing, given that this had only helped the performance previously, we decided to retain the full baseline feature set plus the addition of word vectors.

5.4 Encoding

In addition to BILOU, we also tested BIO and IO encoding schemes with the same feature set discussed in the previous section: the full set of baseline features, plus 1.0-scaled word vectors trained on the full corpus. Neither BIO or IO performed as well as BILOU, though the F1 was only slightly lower in each case: 80.52 for BIO and 80.81 for IO.

5.5 Learning algorithm

Finally, we experimented with different learning algorithms within the crfsuite library. In addition to averaged perceptron, which was the baseline learning algorithm used during the above investigations, we examined: passive aggressive, adaptive regularization of weight vector, stochastic gradient descent with L2 regularization, and gradient descent using L-BFGS. As can be seen in Table 4, all but adaptive regularization of weight vector yielded F1 values close to the highest F1 obtained for averaged perceptron during experimentation with the feature set, but none surpassed AP's performance. The model trained using adaptive regularization of weight vector performed relatively poorly, scoring at 70.96 on F1, around ten points lower than the others. Based on these results, we decided to stick with averaged perceptron.

6 Test set results

The final results can be seen in Table 6. The model trained and tuned exclusively on transcript data did quite well on fan fiction data, with an F1 of 77.04 on the test set in comparison to the F1 of 81.38 on the dev set. Looking at the F1 scores by entity type, we can see that lowest-performing entity category

Type	Prec	Rec	F1
ALL	82.99	79.81	81.37
GPE	69.57	64	66.67
LOC	75.61	72.73	74.14
ORG	75.86	56.90	65.02
PER	88.92	88.02	88.47
SHIP	77.44	72.03	74.64
SPEC	78.07	74.17	76.07

Table 3: Passive Aggressive

Type	Prec	Rec	F1
ALL	81.58	76.67	79.05
GPE	61.70	58.00	59.79
LOC	74.17	69.45	71.74
ORG	73.56	55.17	63.05
PER	86.62	85.68	86.15
SHIP	81.08	62.94	70.87
SPEC	79.44	70.83	74.89

Table 4: Gradient descent using L-BFGS

Type	Prec	Rec	F1
ALL	73.64	68.47	70.96
GPE	42.11	48	44.86
LOC	63.17	62.36	62.76
ORG	50.85	51.72	51.28
PER	84.33	77.05	80.53
SHIP	62.81	53.15	57.58
SPEC	72.77	61.25	66.52

Table 5: Adaptive Regularization of Weight Vector

Type	Prec	Rec	F1
ALL	83.9	871.16	77.04
GPE	69.39	66.6	68
LOC	49.68	39.95	44.29
ORG	71.81	50.23	59.12
PER	89.26	76.54	82.41
SHIP	81.03	62.88	70.81
SPEC	88.13	81.59	84.74

Table 6: Final results on the test set

on the test set was not ORG, but LOC, with an F1 of 44.29. In comparison, LOC performed at an F1 of 73.51 on the dev set. There may be a very good reason for this, and one that could explain part of the slight decrease in performance from dev to test – the transcripts contain location tags that precede every screen, and these are always surrounded by square brackets. For example, "[The bridge]" will directly preceding a scene which takes place in the bridge. We made the decision to annotate these, in addition to other stage instructions, and the model likely learned to rely on the presence of surrounding square brackets to help it to identify an entity as a LOC. Given that this type of description does not appear in fan fiction, which are only narrative prose, it should not be surprising that this category suffered. A more surprising result was that our weakest-performing entity on the dev set, ORG, actually performed slightly higher on the test set, with F1 of 68 in comparison to 63.04 on the dev set.

7 Discussion

Overall, we considered the F1 results on the test set to be successful, in particular because we were not sure what we could expect from a model trained in a significantly different domain. However, there were some shortcomings that we became aware of both during the annotation process and afterwards that we will discuss here.

First, the inconsistency and general sparsity of the tag GPE became problematic; this was the entity type that performed the worst during dev. This was not surprising, given the nature of the *Star Trek* universe: geopolitical entities (GPEs) tend to occupy entire planets (LOCs), and, moreover, consist of a single species (SPEC). Additionally, GPEs are interesting entities in that they, by definition, consist of people (PER), locations (LOC), and organizations (ORG), and therefore can be difficult to pin down even in the best of circumstances. (Some guidelines will have annotators tag GPE-LOC, GPE-PER, etc.) When writing the guidelines, we suspected that all these would combine and make GPE very challenging to tag consistently.

	PER		LOC		
A	1449	54.56%	603	22.70%	
B	1452	56.63%	532	20.75%	
C	1599	52.98%	748	24.78%	
D	1453	53.62%	650	23.99%	
All	6177	54.03%	2679	23.43%	
	GPE		SPEC		
A	66	2.48%	259	9.75%	
B	54	2.11%	269	10.49%	
C	58	1.92%	250	8.28%	
D	2	0.07%	277	10.22%	
All	193	1.69%	1107	9.68%	
	SHIP		ORG		Total
A	146	5.50%	133	5.01%	2656
B	154	6.01%	103	4.02%	2564
C	210	6.96%	153	5.07%	3018
D	148	5.46%	180	6.64%	2710
All	689	5.46%	587	5.13%	11432

Table 7: Relative frequencies of labels by annotator

To mitigate these issues, we prioritized inter-annotator agreement and designed our guidelines to strongly prefer SPEC in the cases of geopolitical ambiguity between these two categories, which left GPE only for the few cases in which a geopolitical entity was given an explicitly geopolitical name, such as Klingon Empire, or was not tied to a single culture or species, such as the Federation. This leaves open the question of whether GPE should have been included at all, and what, if any, value it did bring to the annotation corpus. The trade off for not including GPE would be, of course, that those entities which did fall smartly into this category (Klingon Empire, for example) might not be annotated at all. However, that was already the case for quite a few entity types which we did not set out a tag for.

For the most part, there likely can be no answer without a specific downstream task for the NER, which this investigation did not have. We set out to push the limits of our model, and including GPE was a challenge that we definitely have room to improve on in further work. If we were to continue work on this model, our first task would be to do another two or three rounds of calculating IAA and

adjusting the guidelines to make sure our annotators have the same understanding of each tag. We suspect that a lot of confusion, if not all of it, could be mitigated this way.

Next, as discussed briefly in the results above, the decision to annotated stage directions in the transcripts, including the 'location' cues within square brackets, was made at a very early stage in the annotation guidelines in conjunction with the decision not to annotate the character speaking cues (i.e., 'PICARD:'). The logic behind not annotating dialogue cues is that they are extremely systematic and are always PER entities (in other words, they could be removed or tagged with a simple script using regexes, and don't need language understanding to recognize). Our reason for including the location cues while not including dialogue cues is that stage directions are stylistically more diverse. Sometimes they are just locations ("[Captain's quarters]"), which might be better off unannotated. Sometimes they are stage directions, like "[Chakotay stands up to give the drink to Seven of Nine]", which is a little more ambiguous as to its relevance. Sometimes they are notes from the transcriber like "[Say "Hi!" to DeForest Kelley in oldface makeup!]". However, given the poor performance of the LOC category in the final results on the fiction, more discussion on how to deal with these labels is necessary for future work.

Finally, we'll discuss some smaller areas where the model could be improved, and future extensions of the project. The word embeddings and Brown clusters could be a powerhouse to help the performance improve, as we've seen in other circumstances. Our embeddings were only trained on a relatively small data set (only around 23k sentences). We are interested to see what sort of improvements could be made if they were trained on more data. We would also be curious to see if training on fan fiction and testing on scripts would give significantly different results.

Acknowledgments

We would like to thank Constantine Lignos for an awesome semester.

References

LDC, 2008, Automatic Content Extraction [www.ldc.upenn.edu/Projects/ACE/]

8 Annotation Guidelines

A General Guidelines

1. Do not annotate whitespace outside of the entity (on either side of the beginning/end).

2. Dont include punctuation unless obviously part of the entity.

3. Do not annotate entities inside of other entities.

4. Be greedy; always take the biggest possible entity you can.

   In Starfleet Academy, tag Starfleet Academy and not Starfleet.

5. Annotate titles along with names whenever they occur (E.g. in the sentence Captain Picard walked onto the bridge, annotate Captain Picard as an entity).

6. Do not annotate articles/determiners as parts of entities (The Enterprise should be annotated as Enterprise, for example.)

   A Romulan warbird crash landed on Earth –Tag Romulan warbird without A

7. Annotate stage directions as normal.

8. Dont annotate dialogue tags as PER (e.g. KIRK: Bones, I'm a busy man. –Dont tag KIRK

9. Do not annotate DESCRIPTIONS or other pointers. For instance:

   (a) Generic location stage settings like [Forest] or [Camp] are not LOC

   (b) Multiword descriptions pointing to specific entities, which may or may not contain named entities, are not entities

   Federation colony on Beta Agnii Two – This contains two entities (Federation, GPE, and Beta Agnii Two, LOC) but is not itself LOC or any entity

   The oldest planet in the star system

   My only yeoman

   (c) Pronouns – these are not PER

   (d) Nicknames or forms of address are not descriptions – they are treated as names and are annotated

   (e) I.e., The Captain when referred to a specific individual who happens to be captain in third person, or Captain when addressing said individual Contrast this with a sentence such as The captain of the Romulan ship beamed aboard, in which the captain (referring to an as-yet unspecified individual) is a mere descriptor

B Entity Types

B.1 PER

1. Tag names, nicknames, and titles of people when they refer to a specific individual

2. A note on titles:

   Only tag individuals who are referred to by name or title. Do not tag descriptions of / pointers to individuals.

   The captain_PER was here. He told me there were only Klingons on the ship.

   Captain_PER, theres a message for you!

   My only yeoman and two others dead, seven injured. –Do not tag anything here.

   Do NOT tag titles when referring generically to a position, such as Oh, youll make captain one day!! or This ship really needs a captain

3. A note on extent:

   Do include titles when titles appear near a persons name

   Ensign Ro [Ensign Ro]_PER

   Lieutenant Commander Data [Lieutenant Commander Data]_PER

   Do not tag descriptive content that is part of the NP but isnt a title/name

   The highly logical Spock [Spock]_PER

   The half-human Worf half-[human]_SPEC, [Worf]_PER

   The Captain of the Enterprise, James Kirk [James Kirk]_PER

   Captain Christopher Pike, United Space Ship Enterprise. [Captain Christopher Pike]_PER, [United Space Ship Enterprise]_SHIP

4. Tag nicknames (like Bones).

5. Do NOT tag pronouns

B.2 LOC

1. Specific places, planets, rooms in the ship, areas of space, solar system

2. Starbases are included here and not in SHIP since they are basically stationary and operate more like planets than ships.

3. Examples: Starbase Eleven, Talos Four, Neutral Zone, holodeck, Beta Renner system

4. Includes ship locations: Transporter Control, Medical, engine room, Ten-Forward, In the Talos star group, annotate Talos star group since star group is a crucial part of the definition of the LOC.

5. When Engineering refers to an area of the ship (e.g. Theres a hull breech in Engineering!) annotate as LOC. When Engineering refers to the group of people working there (e.g. I just got transferred to Engineering, Shes the most accomplished officer in Engineering, Engineering to Sick Bay), annotate as ORG.

6. If a location is described generically, such as a very crowded asteroid belt or the planet, do not annotate this (only annotate specific references to specific locations)

B.3 SHIP

1. Manmade space-faring vehicles/places to live which can move through space.

2. Shuttlecraft are included here, as well as war ships.

3. Starbases are not included here since they are essentially unmoving and operate more as planets than ships. (Starbases are considered LOC.)

4. Examples: Enterprise, Romulan warbird

5. When the name of a ship is preceded by a generic descriptor of the craft type, tag only the (Proper) name of the ship – i.e., in the starship Voyager – tag only Voyager.

B.4 GPE

1. A composite entity defined by having a population and a government.

2. The definition of GPE is complicated slightly in this domain since many groups we would normally consider governing bodies tend to have a less direct relationship with a certain geographic location (as compared with an Earth-centric domain, like real life).

3. Klingon Empire seems like a GPE since it is focused on a certain planet. However, Federation also has a certain area of space which it occupies, and although it is not explicitly aligned with any particular race, it is overwhelmingly Human and headquartered on Earth.

4. Examples: Nazi Germany, Klingon Empire, Federation, Dominion

5. Disambiguation: GPE vs. LOC:

   My actions were taken in the best interest of Vulcan_GPE and the High Command_ORG.

   Annotate planets, star systems, and other geographical entities as LOC by default UNLESS it is clear from context that the referent is functioning explicitly as a people with a government, as in the above example. If its unclear whether the entity is functioning as a GPE, ask yourself: in the context in which it is found, can the entity of focus be said to possess agency or animacy? If the entity would otherwise be a LOC and the answer is no, then it is still a LOC.

B.5 ORG

1. To distinguish it from GPE, ORG may have a government of sorts, but no population whose members are not also members of the government.

2. Examples: Galactic Council, Starfleet, launch bay crew

3. Ship-group: e.g. Engineering, Medical

4. Engineering to Sick Bay and the like are pretty common when someone from one group is calling the other group on the intercom.

B.6 SPEC

1. Tag all explicitly named animal species, humanoid or otherwise

   We taggin cows

2. This can be a noun (the Jarada_SPEC), or an adjective (Vulcan_SPEC monastery)

   the half-Vulcan_SPEC officer Spock_PER

3. Examples:

   We are about to make a brief but necessary contact with the Jarada_SPEC, a reclusive, insect-like race known for its idiosyncratic attitude towards protocol

   Tips: Dont highlight articles – only the species name itself

4. When the species name embedded within the name of an ORG, such as the Vulcan council or some such, do not annotate as SPEC, prefer the larger (greedy) ORG annotation

5. Even if a SPEC is acting in an apparent sociopolitical capacity (i.e., at war with another named SPEC), prefer the SPEC tag over GPE. GPE is reserved for those entities with a clear, explicit government which extends over a population of governed subjects, a level of definition not easily achieved (and which should not be read into the mere mention of any group united by common interest, culture, or genetic similarity).