

Διαλογικά Συστήματα και Φωνητικοί Βοηθοί (M913)

Final Assignment Report

Δήμητρα Κοντοέ It1200028

CINEbot

Ανάπτυξη ενός διαλογικού συστήματος (CINEbot) για αναζήτηση κινηματογραφικών ταινιών και πληροφοριών σχετικά με αυτές με λήψη δεδομένων σε πραγματικό χρόνο (<https://www.themoviedb.org/documentation/api>) και χρήση του Rasa (Open Source Conversational AI Framework).

Περιγραφή της λειτουργικότητας του συστήματος:

- Αναζήτηση ταινιών με βάση τον/την σκηνοθέτη
- Αναζήτηση ταινιών με βάση έναν/μία ή/και περισσότερους ηθοποιούς
- Αναζήτηση την ημερομηνίας κυκλοφορίας μιας ταινίας με βάση τον τίτλο της
- Αναζήτηση της πλοκής μιας ταινίας με βάση τον τίτλο της
- Αναζήτηση της βαθμολογίας μιας ταινίας με βάση τον τίτλο της
- Αναζήτηση δημοφιλών ταινιών με βάση τη χρονολογία

Rasa Components - Αρχιτεκτονική:

-Rasa NLU

Το Rasa NLU (Natural Language Understanding) αναλαμβάνει την αποκωδικοποίηση του μηνύματος εισόδου (intent classification, entity extraction) και λειτουργεί με βάση μοντέλα επιβλεπόμενης μάθησης. Συνεπώς, χρειαζόμαστε δεδομένα (training data) για την εκπαίδευση του NLU μοντέλου. Στα training data, ορίζουμε ουσιαστικά τα intents και τα entities.

- tokenization
- μετατροπή του κειμένου εισόδου σε vectors (Featurizer)
- ταξινόμηση των intents (Intent Classifier)
- εξαγωγή των entities (Entity Extractor)

Η παραμετροποίηση του NLU component ορίζεται στο config.yml αρχείο μέσω του pipeline.

-Dual Intent and Entity Transformer (DIET):

Η αρχιτεκτονική αυτή βασίζεται σε transformer μοντέλο που εκτελεί ταυτόχρονα intent classification και entities recognition. Με τη χρήση ενός CRF (Conditional Random Field) tagging layer πάνω στην ακολουθία εξόδου του transformer προβλέπεται μια ακολουθία entity labels, ένα label για κάθε token της ακολουθίας εισόδου. Η έξοδος του transformer για το ειδικό token "CLS" και για τα intent labels μεταφέρονται σε έναν

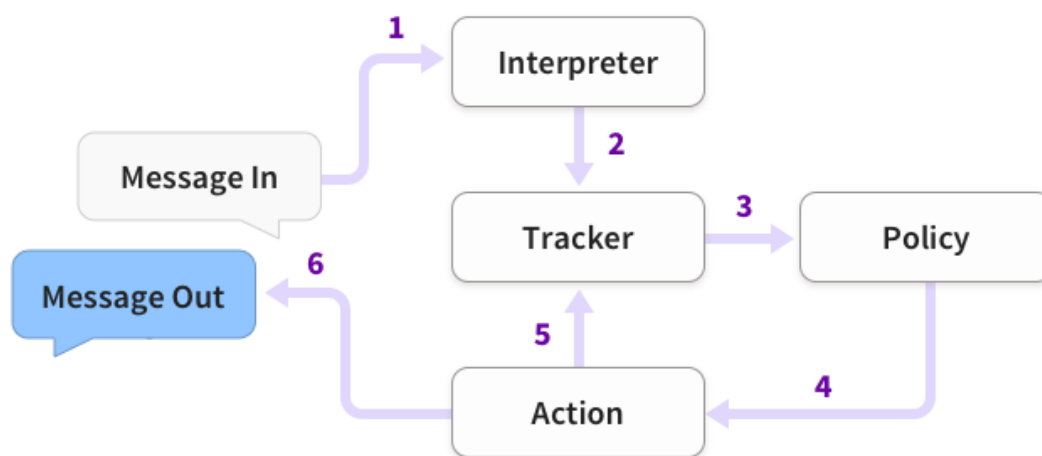
ενιαίο διανυσματικό χώρο και χρησιμοποιείται dot-product loss για τη μεγιστοποίηση της ομοιότητας με τα target label.

Για την ανάπτυξη του cinebot ο DIETClassifier χρησιμοποιείται μόνο για intent classification (entity_recognition: false), καθώς λόγω της χρήσης lookup tables για το entity extraction, με βάση το documentation του rasa, προτείνεται η χρήση του RegexEntityExtractor.

Ο DIETClassifier μπορεί να χρησιμοποιηθεί είτε με dense featurizers είτε με sparse. Στην περίπτωση μας χρησιμοποιήθηκαν οι default: RegexFeaturizer, LexicalSyntacticFeaturizer (sparse, features for entity extraction), CountVectorsFeaturizer (sparse, 1-4grams).

- Dialogue Management

Το στοιχείο Dialogue Management δέχεται σαν είσοδο την έξοδο του Rasa NLU και ουσιαστικά αναλαμβάνει την παραγωγή των απαντήσεων του chatbot με τη χρήση μοντέλων μηχανικής μάθησης. Πιο αναλυτικά (με βάση την παρακάτω εικόνα: <https://rasa.com/docs/rasa/architecture/>) το μήνυμα εισόδου περνάει στον Interpreter για την εξαγωγή του Intent και των Entities. Κατόπιν περνάει στον Tracker ο οποίος αποθηκεύει την τρέχουσα κατάσταση του διαλόγου, η οποία με τη σειρά της περνάει στα μοντέλα Policy (dialogue management models) για να εφαρμοστεί κάποιος αλγόριθμος μηχανικής μάθησης (ή/και rule-based) προκειμένου να επιλεγεί το επόμενο Action, δηλαδή η "απάντηση" του bot. Το Action ενημερώνει τον Tracker με την τρέχουσα κατάσταση του διαλόγου και η απάντηση αποστέλλεται στον χρήστη.



- Policies

Για κάθε απόκριση (turn) σε έναν διάλογο, το κάθε policy που ορίζεται στο config.yml προβλέπει το επόμενο action με κάποιο επίπεδο εμπιστοσύνης. Το policy εκείνο που κάνει την πρόβλεψη με το μεγαλύτερο επίπεδο εμπιστοσύνης αποφασίζει ποιο θα είναι το επόμενο action του bot. Σε περίπτωση που δύο policies κάνουν μια πρόβλεψη με το ίδιο επίπεδο εμπιστοσύνης για την επιλογή του επόμενου action λαμβάνεται υπόψη το policy με το υψηλότερο priority με βάση την default κατάταξη του rasa.

Policy Priority: RulePolicy: 6, MemoizationPolicy: 3, UnexpectEDIntentPolicy: 2, TEDPolicy: 1

Για την ανάπτυξη του CINEbot αξιοποιήθηκε το σύνολο των default policies (MemoizationPolicy, RulePolicy, UnexpectTEDIntentPolicy, TEDPolicy). Για την εκπαίδευση των policies χρησιμοποιείται το αρχείο stories.yml (που περιλαμβάνει τα training examples), καθώς και το domain.yml, όπου ορίζονται τα slots, τα intents, τα entities και τα actions.

-TEDPolicy (Transformer Embedding Dialogue Policy)

Πρόκειται για μια πολιτική βασισμένη στην αρχιτεκτονική transformer, σύμφωνα με την οποία αποφασίζεται σε ποια συγκεκριμένη απόκριση του διαλόγου θα δοθεί προσοχή και ποια θα αγνοηθεί επιλεκτικά για την εξαγωγή της εκάστοτε πρόβλεψης. Σε κάθε turn του διαλόγου, το TEDPolicy λαμβάνει ως είσοδο τριών ειδών πληροφορίες: το μήνυμα του χρήστη, το προηγούμενο action που έχει προβλέψει το σύστημα και τα values που έχουν αποθηκευτεί στη μνήμη του συστήματος για τα slots που μπορεί να υπάρχουν. Οι πληροφορίες αυτές μετατρέπονται σε vectors και τροφοδοτούνται στον transformer μαζί, με τη μορφή πίνακα. Χάρη στον μηχανισμό self-attention, το μοντέλο αποκτά δυναμικά πρόσβαση σε διαφορετικά μέρη του διαλόγου σε κάθε turn και κατόπιν εκτιμά και επανυπολογίζει τη συνάφεια με τα προηγούμενα turns. Έτσι το TED policy μπορεί σε κάποιο turn να λάβει υπόψη του ένα συγκεκριμένο μήνυμα του χρήστη και σε κάποιο άλλο μπορεί να το αγνοήσει τελείως. Κατόπιν, η έξοδος του transformer περνάει από ένα dense layer ώστε να εξαχθούν embeddings που αποτελούν τις αναπαραστάσεις της σημασίας του κειμένου, τόσο του συγκεκριμένου του διαλόγου όσο και των actions. Έτσι υπολογίζεται το loss ώστε το TED policy να μεγιστοποιήσει την ομοιότητα με το target label και να ελαχιστοποιήσει τις ομοιότητες με τα λάθος labels, με βάση τον αλγόριθμο Starspace. Η διαδικασία αυτή της σύγκρισης των ομοιοτήτων μεταξύ των embeddings είναι παρόμοια με εκείνη που εφαρμόζεται για το intent classifications στο Rasa NLU.

-UnexpectTEDIntentPolicy

Πρόκειται για ένα βοηθητικό policy, που βασίζεται στην ίδια αρχιτεκτονική με το TED, και χρησιμοποιείται για την ενεργοποίηση του default action “action_unlikely_intent”, μέσω του οποίας δίνεται η δυνατότητα στο bot να αντιδρά σε περιπτώσεις λιγότερο πιθανών αποκρίσεων του χρήστη.

-RulePolicy

Στην περίπτωση του CINEbot, δεδομένης της απουσίας rules, το συγκεκριμένο Policy χρησιμοποιείται για την ενεργοποίηση των default actions “action_restart” και “action_back”.

-MemoizationPolicy

Το MemoizationPolicy λειτουργεί με τρόπο που διασφαλίζει ότι το bot «θυμάται» τα stories που περιλαμβάνονται στα training data. Ελέγχει εάν η ροή του τρέχοντος διαλόγου αντιστοιχεί με τη ροή κάποιου από τα stories του αρχείου stories.yml. Εφόσον αυτό ισχύει, προβλέπει το επόμενο action με βάση το αντίστοιχο story των training data με επίπεδο εμπιστοσύνης 1,0. Σε αντίθετη περίπτωση, προβλέπει “None”, με επίπεδο εμπιστοσύνης 0.