Είναι ένα paper που έγραψε ο Jens Lehmann που είναι ο επικεφαλής του AKSW. Παρουσιάζει τον αλγόριθμο CELOE που χρησιμοποιείται ως ένας από τους βασικούς αλγορίθμους του DLLearner. To paper το είδα στα πλαίσια της διπλωματικής του Στέλιου Βέρρου ([https://github.com/vrstelios/MachineLearningInKnowledgeGraphs](https://github.com/vrstelios/MachineLearningInKnowledgeGraphs" \t "_new)).

Η λογική είναι ότι ο αλγόριθμος CELOE προσπαθεί να "μάθει" κλάσεις (class expressions) με βάση θετικά και αρνητικά controls. Πχ o στόχος του αλγόριθμου είναι να "μάθει" την περιγραφή της κλάσης "Father" αν του δώσει κάποιος ως input την οντολογία father.owl όπως αυτή περιγράφεται στο manual του DL Learner ([https://github.com/vrstelios/MachineLearningInKnowledgeGraphs/blob/master/manual%20dl-learner.pdf](https://github.com/vrstelios/MachineLearningInKnowledgeGraphs/blob/master/manual%20dl-learner.pdf" \t "_new)).

Αν ανοίξει κάποιος το father.owl θα δει ότι η κλάση father δεν ορίζεται. Αντίθετα, περιγράφονται κάποια θετικά controls και κάποια αρνητικά controls – το παράδειγμα αυτό δεν είναι από το paper, είναι από το manual. Αντίστοιχα, στο παράδειγμα του paper αυτού, η κλάση "Capital" – αυτή είναι η κλάση που προσπαθούμε να μάθουμε – ορίζεται. Παρόλα αυτά, το σύστημα προσπαθεί να μάθει ιδιότητες της κλάσης οι οποίες δεν ορίζονται αρχικά.

Πχ το ότι μια πρωτεύουσα (Capital) θα πρέπει να ικανοποιεί και τις εξής σχέσεις (OWL expressions):

 City and isCapitalOf at least one GeopoliticalRegion

 City and isCapitalOf at least one GeopoliticalRegion

Όπως περιγράφεται στο paper, το πρόβλημα της εκμάθησης ανάγεται στο εξής:

* Έστω ότι θέλουμε να μάθουμε την κλάση "Α" η οποία έχει κάποια individuals/instances στην οντολογία/Knowledge Base K.
* Αν έχω μια περιγραφή (OWL expression) C της κλάσης Α χρησιμοποιώντας τα αξιώματα "A" [= "C" (δηλαδή η A είναι υποσύνολο της C)] ή A=C (δηλαδή η Α ταυτίζεται με την έκφραση C), τότε από τη διαδικασία της εκμάθησης μπορεί να προκύψουν και άλλα αξιώματα πέραν από αυτά με τα οποία η κλάση A ορίζεται αρχικώς.
* Ο στόχος είναι να προκύψει ένας τελεστής – στον οποίο αναφερόμαστε ως retrieval reasoner – ο οποίος συμβολίζεται ως R(C) και όταν εφαρμοστεί στη βάση γνώσης K, επιστρέφει όλα τα instances που ικανοποιούν την έκφραση της C.
* Συνεπώς, ο στόχος είναι να "μάθουμε" έναν τελεστή R και μια έκφραση C σύμφωνα με την οποία R(C) = R(A).

Προφανώς, η έκφραση C την οποία μάθαμε αποτυπώνει μια περιγραφή των instances της Α. Συνεπώς, αν μπορέσουμε να "μάθουμε" μια τέτοια περιγραφή C, τότε μπορούμε στη βάση γνώσης K να προσθέσουμε ένα αξίωμα "A" [= "C"] ή A=C.

Ένας τέτοιος αλγόριθμος εκμάθησης χαρακτηρίζεται ως "πλήρης" αν μπορεί να εγγυηθεί ότι θα βρει "λύση", αν μπορεί δηλαδή να εγγυηθεί ότι θα βρει μια έκφραση C η οποία περιγράφει τα individuals/instances της κλάσης Α.

**[Διαβάζοντας αυτό, μου γεννήθηκε ένα ερώτημα:]**  
Υπάρχει η περίπτωση τα individuals μιας κλάσης A να μπορούν να περιγραφούν με παραπάνω από μια εκφράσεις – πχ την C1, τη C2 κλπ. Αυτές οι εκφράσεις μπορούν να συνυπάρχουν ή/και να είναι αντικρουόμενες. Σε αυτή την περίπτωση, μάλλον δεν είναι σαφές τι γίνεται. Ο αλγόριθμος υποθέτω ότι θα καταλήξει σε μία από όλες τις C – τοπικό ακρότατο – και θα αγνοήσει τις υπόλοιπες.

Εάν μιλούσαμε για reasoner που βασίζεται στο open-world assumption, εκεί τα πράγματα θα ήταν ακόμα πιο δύσκολα πιστεύω.

Σύμφωνα με τη λογική του συγγραφέα, θα πρέπει οι "λύσεις" του αλγορίθμου (δηλαδή οι εκφράσεις της κλάσης C) να είναι όσο το δυνατόν πιο απλές. Την "απλότητα" την ορίζει με τρόπο κατά τη γνώμη μου πολύ απλοϊκό – σχεδόν τσαπατσούλικο – με βάση το πόσο "μακριά"/"μεγάλη" είναι μια έκφραση (length).

Η "ακρίβεια" ορίζεται ως η ικανότητα ορθής κατηγοριοποίησης των instances που εντάσσονται στην οντολογία μετά την εκπαίδευση/εύρεση της έκφρασης C.

Αν έχω μια έκφραση C με μεγάλη "κάλυψη"/ευστοχία στο σύνολο της εκπαίδευσης και μικρή "ακρίβεια", ουσιαστικά έχω "overfit" στο συγκεκριμένο σύνολο εκπαίδευσης. Το πόσο "απλή" είναι μια λύση μπορεί να επηρεάσει σημαντικά την ακρίβειά της – που είναι και το τελικό ζητούμενο.

Συνεπώς, το γεγονός ότι ο αλγόριθμος έχει πολύ έντονο bias προς τις "μικρές"/"απλές" λύσεις δεν είναι πάντα καλό. Θέλει ερμηνεία.

Γενικώς, κατά τη γνώμη μου, θα πρέπει τα expressions C που προκύπτουν κατά τη διαδικασία της εκμάθησης να προσπαθούμε να τα ερμηνεύσουμε. Αν βρίσκουμε κλάσεις/εκφράσεις C οι οποίες δεν αντιστοιχούν σε έννοιες του φυσικού κόσμου, ενδεχομένως κάτι να μην πηγαίνει καλά.

Το αντιφατικό του πράγματος είναι ότι αν ξέραμε να ερμηνεύσουμε τις κλάσεις C, τότε (μάλλον) θα ήταν ήδη καταγεγραμμένες στην οντολογία.

Πχ αν ξέραμε ότι έχει νόημα η έννοια "father", μάλλον θα την είχαμε μοντελοποιήσει ως κλάση και δεν θα περιμέναμε από έναν αλγόριθμο να την ορίσει ως έκφραση για να τη μάθουμε.

Αυτό σημαίνει, κατά τη γνώμη μου, ότι η διαδικασία της εκμάθησης καινούριων κλάσεων/εκφράσεων μπορεί να αξιολογηθεί μόνο μέσω ανθρώπων/domain experts.

Συνεπώς, αυτό σημαίνει ότι οι διαδικασίες αυτές – εκπαίδευση και αξιολόγηση από άνθρωπο για κάθε έννοια – προχωράνε χέρι-χέρι.

Ο αλγόριθμος CELOE παρουσιάζεται στο Figure 1 και είναι αλγόριθμος που ακολουθεί τη λογική "generate and test".

Ουσιαστικά, λειτουργεί ως εξής:

1. Από ένα ή/και περισσότερα positive controls αντλούνται κάποιες εκφράσεις και σχηματίζονται μια ή/και περισσότερες υποψήφιες εκφράσεις C.
2. Κάθε υποψήφια C αξιολογείται με βάση όλα τα positive και τα negative controls.
3. Οι εκφράσεις αυτές ταξινομούνται ανάλογα με τον βαθμό "κάλυψης" και τον βαθμό "ακρίβειάς" τους.

Σημειώνω ότι το paper δεν έχει έντονη μαθηματική κατεύθυνση – εξ όσων μπορώ να καταλάβω.

Στο paper [24] "J. Lehmann, P. Hitzler, Foundations of refinement operators for description logics, in: ILP 2007, vol. 4894 of LNCS, Springer, 2008, best Student Paper Award" περιγράφονται τα "refinement operators" για τα description logics.

Ουσιαστικά, αυτό αποτελεί το μαθηματικό υπόβαθρο του αλγόριθμου CELOE.