

A Survey and Classification of Controlled Natural Languages (2013)

Una classifica di 100 linguaggi CNL (Controlled Natural Language) basati sull'inglese.

In generale i CNL si imparano imparando a NON usare certi costrutti ambigui del linguaggio naturale.

CNL

CNL è linguaggio tale che:

- Basato su lingua naturale.
- Restrizioni lessicali, sintattiche o semantiche.
- Parlanti possono capirlo intuitivamente.
- Costruito a tavolino.

Categorie

Sovrapponibili.

- C comprensibilità
- T traduzione
- F formalismo
- W scritto
- S parlato
- D domain-specific
- A accademico
- I industriale
- G governativo

Classifica PENS

Ciascuna di queste metriche è divisa in 5 livelli.

1. Precisione

Livello di indipendenza dal contesto.

2. Espressività

La dimensione dell'insieme di proposizioni esprimibili.

3. Naturalezza

Livello di somiglianza ad una lingua naturale.

4. Semplicità

Semplicità implementativa del linguaggio.

Punti Salienti

- Pochi linguaggi CNL sono solo parlati.
- Correlazioni abbastanza ovvie (es: Espressività proporzionale $1/\text{Semplicità}$)

Natural language programming: Styles, strategies, and contrasts (1981)

Uno studio empirico con dei soggetti senza alcuna esperienza in programmazione, a cui è stato chiesto di scrivere una specifica in linguaggio naturale per manipolare dei file strutturati.

Punti Salienti

- I problemi sono: stile, semantica e conoscenza pragmatica del mondo.
- I soggetti preferivano gestire le strutture dati in maniera più astratta (“in massa”) rispetto a gestirle con l’iterazione (ciclo for).
- I soggetti si aspettavano che il computer capisse le implicazioni semantiche dei comandi che davano.
- I soggetti facevano uso pesante di riferimenti contestuali.
- I soggetti usavano un lessico abbastanza ristretto (vantaggio).
- I soggetti non controllavano né il formato né la sintassi dei file.
- I soggetti non facevano riferimento a strutture di controllo esplicite (if-then-else, goto, while, for . . .), probabilmente sperando di attingere ad un’esperienza procedurale pragmatica del proprio “interlocutore”.
- I soggetti non assegnavano esplicitamente valori alle variabili, e non le dichiaravano, e usavano sinonimi.
- I soggetti partivano dal passo più cruciale della procedura, per poi precisare quando NON farlo. Al contrario, nei linguaggi di programmazione si ricorre spesso alle guard clauses, o agli if annidati, e si mette il passo cruciale alla fine, o in mezzo.

Program Synthesis using Natural Language (2016)

Una metodologia per la creazione di Domain Specific Languages (DSL) ad hoc, con un'interfaccia utente naturale Natural Language (NL).

Punti Salienti

- Si parte da specifica DSL e specifica NL.
- Vengono forniti esempi di mappatura da NL a DSL.
- Un modello di Machine Learning (ML) è trainato per mappare frasi in NL a statement in DSL.
- Simile a Programming By Example (PBE).
- All'utente finale che inserisce una frase in NL viene offerta più di una alternativa, ordinata per rilevanza, di programma DSL equivalente.

Learning to Transform Natural to Formal Languages (2005)

Metodo per mappare frasi in linguaggio naturali a statements in un linguaggio formale. Approccio di trasformazione incrementale e bottom-up.

Inform 7

Un linguaggio per sviluppare contenuti di “interactive fiction”: una specie di “videogioco”, che permette di interagire, tramite un prompt testuale, con gli oggetti e le stanze della storia.

Il linguaggio somiglia all'inglese, ed è molto orientato verso lo stile dichiarativo. Permette di creare ontologie.

- https://en.wikipedia.org/wiki/Inform#Inform_7
- <https://ganelson.github.io/inform-website/doc/>
- <https://ganelson.github.io/inform-website/downloads/>
- <https://www.gamedeveloper.com/programming/learning-inform-7-part-1>

Programming by Voice, Efficiency in the Reactive and Imperative Paradigm (2021)

Confronto del paradigma imperativo e del paradigma reattivo nell'ambito della programmazione vocale.

Programmazione Reattiva

Incentrata attorno al flusso di dati e alla propagazione dei cambiamenti. Si occupa dei flussi di dati asincroni, tramite l'utilizzo del pattern Observer.

Punti Salienti

- Con paradigma imperativo si devono inserire più parole “innaturali” che devono essere dettate lettera per lettera.
- Con paradigma reattivo c'era una percentuale più alta di parole “naturali”, riconoscibili senza doverle dettare lettera per lettera.
- Si è usato Talon come ambiente di sviluppo vocale.

Talon

Ambiente di sviluppo **closed-source** :’-(mutlimodale, che supporta interazioni vocali (linguistiche e paralinguistiche), eye-tracking ed estensibilità tramite script in Python.

On the Positive Effect of Reactive Programming on Software Comprehension: An Empirical Study (2017)

Punti Salienti

1. La comprensione dei programmi reattivi è meno correlata con il livello di conoscenza in programmazione, rispetto alla comprensione dei programmi orientati agli oggetti.
2. Possibilmente perché si tende a usare meno codice boilerplate.

A Controlled Language for Knowledge Formulation on the Semantic Web (2002)

Clear Talk, linguaggio CNL basato sull'inglese.

Punti Salienti

- Poche ore per imparare a leggerlo, qualche giorno per scriverlo.
- Autore può scegliere livello di ambiguità.
- Disambiguazione sintattica tramite parentesi e altri simboli ASCII.
- Applicazioni relative ad apprendimento (umano) e motori di ricerca semantici.

An Empirical Approach for the Evaluation of Voice User Interfaces (2010)

Bibliography

Tracer Bullets

- <https://www.cin.ufpe.br/~cavmj/104The%20Pragmatic%20Programmer,%20From%20Journeyman%20To%20Master%20-%20Andrew%20Hunt,%20David%20Thomas%20-%20Addison%20Wesley%20-%201999.pdf>
- offline

SWORIER Paper

- <https://arxiv.org/pdf/0711.3419.pdf>
- offline

SWORIER Dealing with Contraddictions

- <https://arxiv.org/pdf/0711.3419.pdf>
- offline

Limitations on Prolog Syntax SWORIER

- <https://arxiv.org/pdf/0711.3419.pdf>
- offline

Anaphora Resolution Algorithm

Based on a theme-rheme (aka: Topic & Comment) subdivision of a sentence. Entities in theme should be looked up omitting their relation to entities in rheme. And viceversa, entities in rheme should be looked up omitting their relation to entities in theme. This ensures that only “old” information is used to look up potential anaphoric references. The nexus between theme and rheme entities is new information, and should be omitted in the lookup.

Theme-Rheme

https://en.wikipedia.org/wiki/Topic_and_comment

React Reconciliation

<https://reactjs.org/docs/reconciliation.html>

Beyond AOP

- <https://www2.ccs.neu.edu/research/demeter/papers/oopsla-onward/beyondAOP.pdf>
- offline

Natural Java Paper

- <https://www.cs.utah.edu/~riloff/pdfs/iui2000.pdf>
- offline

Ontology

- https://protege.stanford.edu/publications/ontology_development/ontology101.pdf
- offline

Decorator Pattern

- <https://it.wikipedia.org/wiki/Decorator>
- https://sourcemaking.com/design_patterns/decorator

Lisperator

<https://lisperator.net/pltut/eval1/>

Bootstrapping

[https://en.wikipedia.org/wiki/Bootstrapping_\(compilers\)](https://en.wikipedia.org/wiki/Bootstrapping_(compilers))

Tau Prolog

<http://tau-prolog.org/>

Facade SW Design Pattern

https://en.wikipedia.org/wiki/Facade_pattern

A Survey and Classification of Controlled Natural Languages (2013)

offline

Natural language programming: Styles, strategies, and contrasts (1981)

offline

Program Synthesis using Natural Language (2016)

offline

Learning to Transform Natural to Formal Languages (2005)

offline

Inform 7

offline

Programming by Voice, Efficiency in the Reactive and Imperative Paradigm (2021)

offline

Talon Voice

- <https://talonvoice.com/>

On the Positive Effect of Reactive Programming on Software Comprehension: An Empirical Study (2017)

offline

A Controlled Language for Knowledge Formulation on the Semantic Web (2002)

offline

An Empirical Approach for the Evaluation of Voice User Interfaces (2010)

offline